NATIONAL COMPUTER SECURITY CENTER

AD-A234 058

FINAL EVALUATION REPORT

OF

UNISYS CORPORATION

OS 1100

DTIC
ELECTE
S APR 0 8 1991
B D

September 27, 1989

DTIC FILE COPY

91 4 05 035

FINAL EVALUATION REPORT

UNISYS CORPORATION

OS 1100

NATIONAL COMPUTER SECURITY CENTER
9800 Savage Road
Fort George G. Meade Maryland 20755-6000

27 September 1989

This page intentionally left blank.

FOREWORD

This publication, the Final Evaluation Report on Unisys Corporation OS 1100, is being issued by the National Computer Security Center under the authority of and in accordance with DoD Directive 5215.1, "Computer Security Evaluation Center". The purpose of this report is to document the results of the formal evaluation of Unisys' OS 1100 system. The requirements stated in this report are taken from *DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA* dated December 1985.

Approved:

[signature] September 27, 1989

Thomas R. Malarkey
Deputy Chief,
Office of Product Evaluations
and Technical Guidelines
National Computer Security Center

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

ACKNOWLEDGEMENTS

Team Members

Team members included the following individuals:

Donald G. Crossman

National Computer Security Center
Fort George G. Meade, Maryland

James Donndelinger
Jeffrey Jones

Aerospace Corporation
Los Angeles, California

Edward J. Coyne
Robert Williamson

MITRE Corporation
McLean, Virginia

CONTENTS

This page intentionally left blank.

EXECUTIVE SUMMARY

The security protection provided by the Unisys OS 1100 operating system, configured according to the most secure manner described in the Trusted Facility Manual [42], running on the 1100/90, the 2200/200, or the System 11 as described on page 6, "Hardware Overview", has been examined by the National Computer Security Center (NCSC). The security features of OS 1100 were examined and tested against the requirements specified by the *DoD Trusted Computer System Evaluation Criteria* (the Criteria) dated December 1985 in order to establish a rating.

This report documents the evaluation team's understanding of the security-relevant features of OS 1100. Based on this understanding, the evaluation team has determined OS 1100 to be Class B1.

Security testing was performed by the vendor and the team. It has been determined that the security features are implemented as designed. All security-related documentation for the system has been reviewed and determined to be adequate and accurate.

This formal evaluation assigns a Class B1 rating to OS 1100. This class is the highest class at which the system meets all Criteria requirements.

Following the description of OS 1100 are the results of the formal team's assessment of how the system meets the evaluation criteria for Class B1.

This page intentionally left blank.

INTRODUCTION

In August 1984, the National Computer Security Center (NCSC) began a developmental product evaluation of OS 1100, a Unisys Corporation (formerly Sperry Corporation) product [1] This report provides evidence and analysis of the security features and assurances of OS 1100. The report documents the evaluation team's understanding of the product's security design and appraises its functionality and integrity against the Criteria's B Division security requirements. The report documents the evaluation team's findings concerning the system's security features and assurances.

Material for this report was gathered by the NCSC Unisys (Sperry) evaluation team through documentation, interaction with system developers, code review, and testing.

Evaluation Process Overview

The Department of Defense Computer Security Center was established in January 1981 to encourage the widespread availability of trusted computer systems for use by facilities processing classified or other sensitive information. In August 1985 the name of the organization was changed to the National Computer Security Center. In order to assist in assessing the degree of trust one could place in a given computer system, the *DoD Trusted Computer System Evaluation Criteria* was written. The Criteria establishes specific requirements that a computer system must meet in order to achieve a predefined level of trustworthiness. The Criteria levels are arranged hierarchically into four major divisions of protection, each with certain security-relevant characteristics. These divisions are in turn subdivided into classes. To determine the division and class at which all requirements are met by a system, the system must be evaluated against the Criteria by an NCSC evaluation team.

The NCSC performs evaluations of computer products in varying stages of development from initial design to those that are commercially available. Product evaluations consist of a developmental phase and a formal phase. All evaluations begin with the developmental phase. The primary thrust of the developmental phase is an in-depth examination of a manufacturer's design for either a new trusted product or for security enhancements to an existing product. Since the developmental phase is based on design documentation and information supplied by the industry source, it involves no "hands on" use of the system. The developmental phase results in the production of an Initial Product Assessment Report (IPAR). The IPAR documents the evaluation team's understanding of the system based on the information presented by the vendor. Because the IPAR contains proprietary information, distribution is restricted to the vendor and the NCSC.

Products entering the formal phase must be complete security systems. In addition, the release being evaluated must not undergo any additional development. The formal phase is

---

[1]Some preliminary work was carried out for about two years prior to 1984. OS 1100 UNISYS OS 1100 Security Release I did not exist in its current form in 1984. Both the product and its name have evolved over time.

an analysis of the hardware and software components of a system, all system documentation, and a mapping of the security features and assurances to the Criteria. The analysis performed during the formal phase requires hands on testing (i.e., functional testing and, if applicable, penetration testing). The formal phase results in the production of a final report and an Evaluated Products List entry. The final report is a summary of the evaluation and includes the EPL rating which indicates the final class at which the product successfully met all Criteria requirements in terms of both features and assurances. The final report and EPL entry are made public.

Document Organization

This report consists of five major sections and seven appendices. Section 1 is this introduction. Section 2 provides an overview of the system hardware and software architecture. Section 3 provides a mapping between the requirements specified in the Criteria and the OS 1100 features that fulfill those requirements. Section 4 discusses the assurances provided to the evaluation team that OS 1100 works as described in the design documentation. Section 5 contains a list of evaluator comments. Appendix A (see page A-1, "Evaluated Hardware") identifies specific hardware components and Appendix B (see page B-1, "Evaluated Software") identifies specific software components to which the evaluation applies. Appendix C (see page C-1, "Acronyms") provides a list of acronyms used, Appendix D (see page D-1, "Glossary of Terms") provides a glossary of OS 1100 terms, and Appendix E (see page E-1, "Team Tests") presents descriptions of the team tests applied to OS 1100. Appendix F (see page F-1, "References") provides the references used. Appendix G describes testing.

SYSTEM OVERVIEW

This section begins with a brief description of the history of OS 1100 and a high-level view of the hardware and software necessary to run the system. The remainder of the section describes the security-relevant architecture and mechanisms used in OS 1100. The information presented in this section is based on a review of vendor supplied documentation, interaction with system developers, code review, and testing.

OS 1100 Background and History

Unisys OS 1100 is a mature third generation operating system with its origins in the early 1960s. Both the software and hardware have evolved over this period and have improved both in performance and security features.

Unisys has integrated a comprehensive security system in UNISYS OS 1100 Security Release I. The earlier models of Unisys computer systems do not support UNISYS OS 1100 Security Release I, which uses architectural protection mechanisms to support the security policy. Therefore only certain models in the Unisys product line may be used with the OS 1100 TCB. For a complete list of the evaluated hardware and software see page A-1, "Evaluated Hardware" and page B-1, "Evaluated Software".

Functional Overview

OS 1100 provides a number of system functions. The 1100 Executive functions can be categorized as:

1. Central Executive - provides control of the hardware-oriented functions, including initialization, interrupt processing, I/O and clock/timer support.

2. Processor Management

3. Main Storage Management

4. Job Scheduling

5. Symbiont Device (unit record device) Control

6. Transaction Processing

7. Auditing

8. Tape and File Management

9. Security/Accounting/Quota Control

10. Integrated Recovery Support

11. Other minor functions that are non-security relevant

External Interfaces

Communication between the system and its environment takes place at user and operator interfaces.

User Interfaces

The 1100 Executive[1] supports three basic modes of operation (described in some Unisys documentation as user interfaces): demand (interactive), batch, and transaction processing (TIP).[2]

Demand is an interactive environment where ECL (see page 35, "Executive Interfaces") statements are entered at terminals by users and are used to request services from the 1100 Executive. A demand session begins with a user identifying and authenticating himself to the 1100 Executive and continues with the initiation of a run (page 53, "Access Control").[3]

The batch environment is similar to batch operations in other systems. A predefined sequence of ECL statements (runstream) is submitted to the 1100 Executive as a single unit with no opportunity for dynamic interaction. Output is generally delayed until execution of the entire sequence is completed. Batch runs are card images typically submitted in a file or interactively and contain statements with user-ids, passwords, and security attributes which are used to identify and authenticate the user. The validated attributes will be in effect for the duration of that run.

TIP is a user-oriented interactive transaction-based environment. In this environment, a user signs on and is validated using the same mechanisms as for demand. A user communicates with TIP via messages that may constitute a terminal screen. Each message contains a transaction code which TIP uses to determine the System Security Administrator (SSA) approved, generally site-produced, transaction program to be executed. That program can initiate another transaction program by generating a passoff

---

[1]The Executive is the nucleus of OS 1100.

[2]Some references, e.g., the 1100 System Concepts Student Guide [39], also mention a fourth mode known as real-time. This mode constitutes a mode of execution rather than a mode of operation, since the primary intent is to ensure a high priority for instruction-processor execution. Programs executed in batch or demand mode can execute at real-time priority by executing a privileged Executive Request (see page 35, "Executive Interfaces"). The evaluated system does not support real-time.

[3]After the user has been authenticated, he may enter a subset of console operator commands from his terminal. The commands allowed are controlled by the SSA on a user-id basis.

message and return a message to the user (commonly in a form template to be filled in and transmitted as the next transaction). The TIP environment is optimized for speed and large numbers of users.

Operator Interfaces

On an OS 1100 system running in B1 mode, two operator interfaces are available: the main system console, which communicates with the main-system executive software, and the Network Management Services (NMS) console, which communicates with the front-end processor software.

Operational Roles

Within Unisys OS 1100 several roles are defined for different personnel using the system. These roles are user (end user, application programmers, and system programmers), operators (system, SSP, and network management), and the SSA. Each of these user types has different privileges and responsibilities to fulfill in maintaining a trusted environment. The B1 operator roles supported are OS 1100 console operator and network administrator.

Operator roles are defined by access to operator consoles. The SSA is defined as having the Master Account Number and Master Password, which give him a full set of system privileges and the maximum security level on the system. Normal users are defined to the system without privileges. Some users may be granted privileges by the SSA.

Users

Generally, within a facility using a trusted computer system, the general user has several fundamental responsibilities. These are to safeguard his data, report suspected or actual breaches (e.g., illegal users, misused passwords, damaged programs or data) to the SSA (or his alternate), and to uphold the security policy[1] as stated in the SFUG [91]. Within the Unisys system and its security policy, additional personnel are considered to be users, but they they may possess more capabilities than ordinary users. These are applications and system programmers who may be granted privilege to execute security-relevant Executive Requests (see page 35, "Executive Interfaces").

Operators

The computer operator within a trusted computer system facility acquires responsibilities beyond those for an untrusted user. Not only is the operator required to keep the system up and running, answer user questions, perform all tape/disk mounts and dismounts, but also the operator is required to safeguard the equipment from security hazards. To do this

---

[1]The security policy, in brief, is user-oriented subjects and MAC and DAC on objects. MAC refers to no read up, no write down, and no write up. DAC defaults to owner-only access and is an access-list implementation. Privileges may override the security policy and the use or potential use of privileges is audited.

the operator must first control all access within the facility, including access to the physical hardware. Secondly, within the OS 1100 system the operator must respond appropriately to security violations. This includes notifying the SSA of potential problems, should the operator detect security faults, and when the system is placed in maintenance mode (see page 19, "System Support Processor (SSP)"). The operator will also be responsible for handling output produced in the facility and its distribution. Other normal duties include communicating with OS 1100, initializing the SSP and the system, taking system dumps, monitoring system information, and starting utilities. Operators communicate with the system through designated operator consoles. Operators are required to maintain a paper log which identifies the operator(s) on duty to attach responsibility to operator actions.

System Security Administrator

The SSA is the person responsible for the security at a site and has the ultimate responsibility to ensure a correct implementation of the security policy. At system initialization, the SSA enters his user-id and the master account number into OS 1100. This individual creates and maintains user profiles, the Security Compartment Definition Table (SCDT), the Security Mandatory Component Definition Table (SMDT), and all other parts of the overall security database (see page 29, "Security Database").

By use of these tables, the SSA controls access to the system under his direction. Within OS 1100, the tables listed above are several of the most critical items in the system. Additionally, the SSA has the responsibility for security checking of all imported data, to safeguard all data of any type on the system, and to control the installation of TIP (see page 61, "TIP Session Access") transaction programs.[1]

Hardware Overview

The hardware component of the Class B1 TCB encompasses three members of Unisys Series 1100 and 2200:

- Series 1100: System 11 and 1100/90

- Series 2200: 2200/200

These models contain C-Series instruction processors which provide the protection mechanisms needed to implement the evaluated system. These mechanisms include locks, keys, and gates and are used to protect banks of memory. C-Series processors also provide extended-mode addressing (page 9, "Instruction Processor (IP) and Main Storage"). This permits additional granularity of process isolation and a larger addressing

---

[1]UNISYS documentation uses the abbreviation SSA to mean either the System Security Administrator himself or a sub-administrator with one or more access privileges (see page 63, "Privileges") of the SSA. For example, access to any of the tables in the security database designates a user as a sub-SSA.

capability over basic mode, although extended mode is not fully exploited in the evaluated version of OS 1100. Extended mode addressing is only used within the operating system, in the evaluated systems. User programs will use basic mode addressing only [4].

The 1100/90, System 11, and 2200/200 may have from one to four instruction processors (IP) and one to four (1100/90 only) input/output processors (IOP). In all cases involving multiple IPs, the systems run asynchronously in tightly coupled, symmetric configurations, with all memory shared and a single copy of the operating system. Additional information concerning evaluated configurations is as follows:
System 11 2200/200 1100/90

Component Interconnection	bus	bus	point-to-point
Communications	CMS & DCP, WSCU, CLCU, or ICP	CMS & DCP	CMS & DCP[1]
Input/Output Processors	yes	yes	yes
System Support Processor(s)	yes (integral)	yes (integral)	yes
Channel Types	Disk Controller Block Mux Byte Bus	Disk Controller Block Mux Byte Peripheral Adapter L-Bus Adapter	Word Channel Block Mux

Architectural Protection Mechanisms

The C-Series instruction processor uses access control mechanisms to ensure that only authorized activities are permitted to read, write, or execute a memory bank (see page 9, "Instruction Processor (IP) and Main Storage").

Addressing is carried out by accessing banks which are based on base registers. Basic mode uses four base registers and extended mode uses 32 base registers, 16 of which are for the executive software.

Each bank is defined by a Bank Descriptor (BD) which resides in a Bank Descriptor Table (BDT). A virtual address references a Bank Descriptor Index (BDI) which selects a BD from the BDT. The protection mechanism of keys and locks (see page 10, "Use of Keys and Locks") makes use of information in the BD of a bank.

---

[1]See page C-1, "Acronyms" and page 17, "Communications".

Common banks are shared by multiple users and may be protected by gates. Common banks protected by a gate are called encapsulated common banks. Gates are data structures in banks which are interposed between an executing program and a target bank. When a program attempts to access an encapsulated common bank, it encounters the BD of a gate bank containing the gate protecting the encapsulated common bank. The gate bank is fetched from memory and the gate data structure is accessed.[1] The Executive receives control at this point and checks the MAC and DAC access permissions of the calling program. If MAC and DAC are satisfied, the Executive activates a second gate which provides access to the referenced encapsulated common bank(s). The second gate contains a specification of the processor privilege and memory access privilege necessary to access the target bank.

The encapsulated common bank mechanism is used to implement subsystems (see page 20, "Subsystems").

C-Series Addressing

The evaluated systems support both basic and extended addresssing. Since the C-Series architecture is relatively new, OS 1100 is only beginning to take advantage of extended mode. Future development will take further advantage of the greater granularity of address space isolation and address space size provided by extended mode addressing. However, basic mode provides sufficient support to implement the OS 1100 security policy.

Virtual memory is divided into levels of an address tree. The C-Series architecture provides for an address tree with four levels. The levels correspond to sets of system banks, application banks, program banks, and activity banks. In the evaluated system, there is only one set of application banks defined:

---

[1]In a virtual address, the Level and BDI portions of the address select the correct gate bank and the offset portion selects the correct gate within the gate bank. The starting point for access within a bank is given by an offset value in a virtual address. When a gate is used, the offset is specified in the gate. Once a bank is accessed, all relative addresses are checked to ensure that they are within the limits of the bank.

C-Series Address Tree

Level 0 | System Banks

Level 1 | Application-1 Banks

Level 2 | Program-1 Banks | Program-2 Banks | Program-3 Banks | . . .

Level 3 | Activity-1 Banks | Activity-2 Banks | Activity-3 Banks | . . .
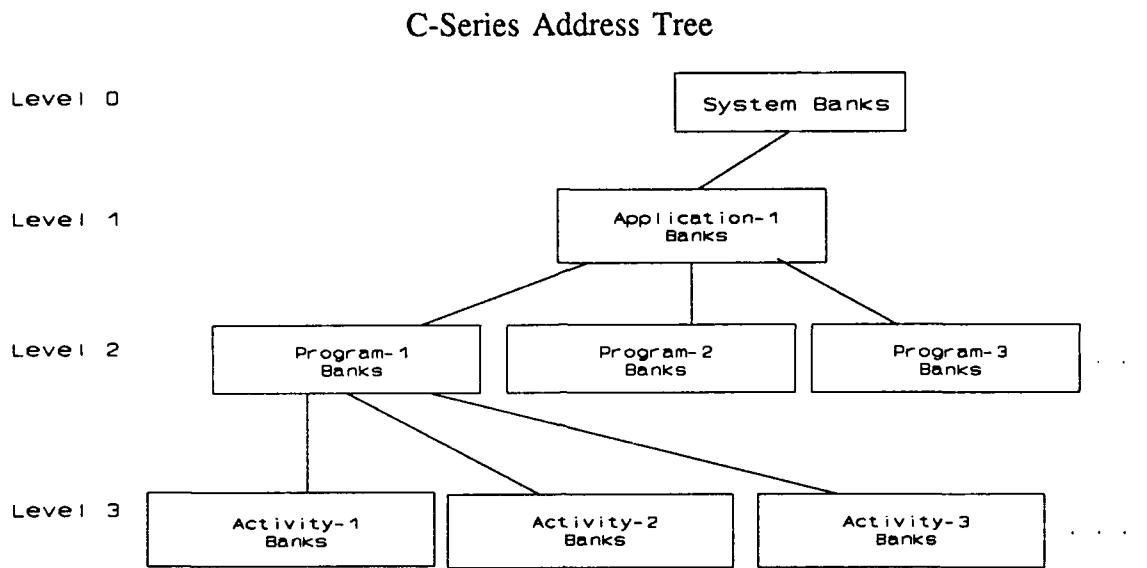
Instruction Processor (IP) and Main Storage

The hardware component of the Class B1 TCB, consisting of the 1100/90, the 2200/200, and the System 11, contains C-Series instruction processors.[1] Multiprocessor configurations are supported. The three models have a common architecture, using different processors and memory, and are built using either point-to-point (1100/90) or bus (System 11 and 2200/200) interconnection. On the 2200/200, memory banks may be paged or unpaged, but this paging mechanism is not used in the evaluated system [4]. The 1100/90 uses processor-local cache memory where cache updates are written through to backing storage.

PROTECTION MECHANISMS

Protection mechanisms are implemented in the instruction processor and main-storage hardware.

---

[1]C-Series processors support both basic and extended mode addressing. Other models in UNISYS' 1100 Series and 2200 Series are not supported by UNISYS OS 1100 Security Release I, the evaluated operating system.

INSTRUCTION PROCESSOR

The C-Series processors support four levels of Processor Privilege, two of which are used in OS 1100:

    0: Executive
    1: Not used or named
    2: User
    3: Not used or named

Privilege values are assigned to each instruction and protected function (e.g., accessing the Executive General Register Set). The instruction or protected function may only be performed when the Processor Privilege value is less than or equal to the assigned value, i.e., the Processor Privilege dominates the assigned value. An Invalid Instruction or Reference Violation interrupt occurs if the Processor Privilege is insufficient [37].

MAIN STORAGE

Memory is physically contained in Main Storage Units (MSU) and logically divided into banks. Access to these memory banks is controlled by using locks, keys and two access permission fields in the hardware. Each memory bank is protected by a lock, in its BD, and each executing process has a key, in the IP's Indicator/Key Register. Two access permission fields exist for a memory bank, one for owner access, called Special Access Permission (SAP), and one for non-owner access, called General Access Permission (GAP). Each of these fields contains enable bits that indicate if read, write, enter or no access is allowed. If the key matches the lock, then the SAP field is used to determine the type of access, if any, that is allowed. If the key does not match the lock, then the GAP field is similarly used. A key value of all zeroes serves as a master key which allows all access types [37].

USE OF KEYS AND LOCKS

The lock and key mechanism is used as follows: Access keys are assigned to activities, i.e., units of work done on an IP. Access locks are assigned to banks. Locks and keys contain two fields, called ring and domain. Matching of a key to a lock (illustrated on page 11, "Matching Keys and Locks") means that either (a) the key and lock domain fields are equal or (b) the key's ring value is numerically less than the lock's ring value.

Areas of virtual address space are isolated or shared by assigning values to the ring and domain fields of locks and keys. Ring values ensure that only processes with sufficient memory privilege may access certain banks. For example, the Executive is protected from unauthorized access by having its banks protected by a lock with a ring value of zero. Locks on encapsulated common banks are formed by assigning the same domain value to all the banks in a common-bank subsystem (page 20, "Subsystems"). The isolation/sharing mechanism for the banks in a given common-bank subsystem is that the domain portion of the lock is the same for all the banks comprising the subsystem and different from the domains of other subsystems. This protects one subsystem from accessing another directly,

while permitting free access among the banks in the subsystem. The Executive assigns a lock's domain value to the set of banks in a subsystem when the subsystem is first installed.

Key   LOCK

| Ring | Domain | | Ring | Domain |

or

<

match   *else*   GAP access

rwe

SAP access   rwe = read, write, enter permission

rwe

Matching Keys and Locks
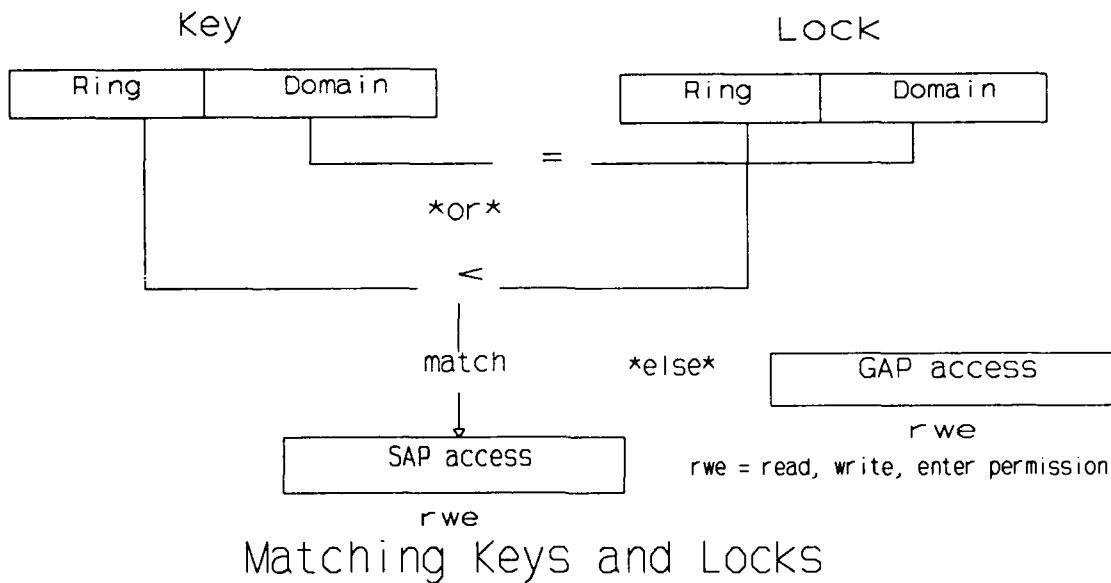
GATES

Gates, described in ASA-0113, C-Series Instruction Processor Architectural Specification [37], provide a mechanism where both memory access privileges and processor privileges may be changed in a controlled and protected fashion without requiring the invoker of the mechanism to have any special privilege. The specification of the privilege changes cannot be made by the invoker. The gate defines where instruction execution is to begin following the privilege changes. These values are established and maintained by the Executive.

A gate is used to control access to a subsystem. Gates are contained in special banks called gate banks. Access to the gate is controlled first by access to the gate bank and then by a lock and two access permission fields contained in the gate. The key, lock, and access permission fields work in the same way as they do for memory bank access, except that only the enter access bit is used in a gate. If access is allowed to the gate, a gate crossing occurs. During the gate crossing, the hardware may change the execution environment as defined by control bits in the gate. The hardware then transfers instruction execution to the virtual address contained in the gate. The parts of the execution environment that can change on a gate crossing are: the access key, processor privilege, and use of the Executive register set.

Addressing Architecture

The OS 1100 TCB uses a base register architecture that also supports indexing as well as indirect and immediate addressing. The C-Series hardware supports the separate addressing modes of basic and extended. The evaluated system uses basic mode for all user mode addressing, while the Executive uses both basic mode and extended mode.

Basic mode consists of an addressing environment that uses four base registers and 262K words of executable address space. Each of these base registers can nominally address a bank of up to 65K words. The hardware uses a pre-established base register preference to determine which base register to use in address generation.

Extended mode supports the use of 32 base registers (16 of these require privileged access) and a greatly expanded user address space. In addition, extended mode supports both 18-bit and 24-bit indexing. A separate field in the instruction word is used to designate the base register to be used in address generation.

The addressing structure has four levels (page 8, "C-Series Addressing"). The addressing and protection mechanisms permit user banks to share access to common banks. C-Series architecture provides the capability of protecting common banks by encapsulating them in subsystems (page 20, "Subsystems"). A common bank subsystem, in brief, is a set of one or more memory banks which are protected by a hardware gate, as well as by keys and locks.

A user's memory banks are always in motion, moving in and out and around main storage. As a result, a given bank's absolute address, in main storage, also changes. The processor manages these address changes with an Instruction Counter (IC) and a set of base registers. The IC contains the next program relative address. The base registers contain the absolute address of each active bank. The effective address is calculated as the sum of the base register and the IC. As the banks move around in main memory, their base registers are changed accordingly. The processor selects a specific base register, for the effective address calculation, with an iterative process that begins with the first base register. The processor compares the program relative address with the base register's upper and lower storage limits. If the program relative address is within range, the base register is selected; otherwise, the same action is repeated for the next register and so on until either a match is found or the process is terminated and charged with a storage limits violation.

System banks are in the address space of all users. Only one application-level set of banks is defined. All users may access this application, which contains the common bank subsystems and common utility routines. At the application level, several applications (e.g., DBMS environments) could exist. The gate mechanism of the subsystem controls access to the applications. Each user's program banks are in the program level. Each program-level bank is isolated from others at its level because a separate Bank Descriptor Table (BDT) is defined for each user. Each BDT points to a different set of banks.

The virtual address space of a program comprises the banks which are addressable by the program. Banks become part of a program's actual address space when they are based. Bank basing is the loading of base registers using bank transfer instructions.[1] Subsystem protection of common banks is achieved through the use of gate banks. The program will encounter a gate bank when it attempts to address an encapsulated common bank. Within the gate bank will be a gate data structure.[2] The gate bank performs two functions for common-bank subsystems:

1. Permit the Executive to intervene and perform access checks.

2. Change the memory and processor privileges from those of the calling subsystem to those of the called subsystem.

In both basic and extended modes, a virtual address contains a level, a Bank Descriptor Index (BDI), and an offset. The level is the address tree level. The BDI is the index into a BDT pointing to a Bank Descriptor (BD). The offset is the point within a target bank where entry will be made. In a common-bank subsystem, the offset at which access will begin is specified not by the virtual address specified by the user's program, but by the gate which is protecting the subsystem.[3] Thus the user's process is completely isolated from direct manipulation of the memory protected by the subsystem.

---

[1]LIJ, LDJ, and LBJ in basic mode and CALL and GOTO in extended mode.

[2]For example, the common bank might be part of a general system service such as the Site Management Complex (SIMAN).

[3]The offset in the user's program merely selects the gate in the gate bank. Banks which are not currently based may be swapped to and from disk (mass storage) as needed.

Process Activation

After selecting a process from the process queue (SWIQUE), the dispatcher must set up the execution environment for the process.

QUEUED PROCESS on SWIQUE

PROCESS CONTROL TABLE (PCT)

```
┌─────────────────────────────────────┐
│  PCT  STRUCTURE  INFORMATION         │
├─────────────────────────────────────┤
│ Control  Information                 │
│ (General  Register  Set)             │
│       -  Designator  registers       │
│       -  Start  Address              │
│       -  instruction  counter)       │
│       -  Quantum  value              │
│       -  Initially  based  banks     │
├─────────────────────────────────────┤
│   BDT                                │
│         Bank  descriptors  for       │
│         each  BDI                    │
└─────────────────────────────────────┘
```
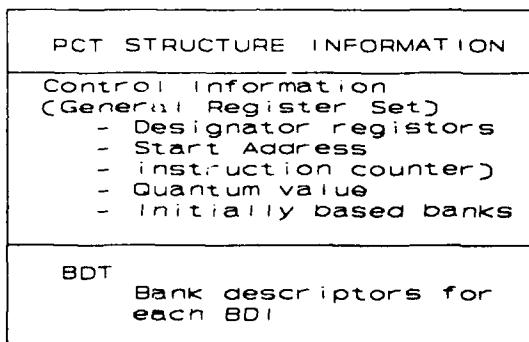
The first step required to establish the execution environment is for the dispatcher to select the process from SWIQUE using a predefined priority scheme. Once the execution environment is established, the dispatcher gives control to the processor through the following steps:

- Initialize the user register stack in the processor.
- Load the base registers with address values and bank attributes from the
  Bank Descriptor Table indexed by a BDI.
- Initialize the designator register.
- Set quantum timer.
- Initialize Instruction Counter with start address.
- Change processor state and give control to the process.

See figure below which illustrates the above steps to activate a process and how the address space of the process is mapped into main storage.

PROCESSOR

MAIN STORAGE

QUEUED PROCESSES

PTC CONTROL INFORMATION
GENERAL REGISTER SET (GRS)
BDT

USER REGISTER STACK
BDI REGISTERS

QUANTUM TIMER
INSTRUCTION COUNTER
DESIGNATOR REGISTER
BASE REGISTERS
UL   BASE ADDRESS
LL   FLAGS
UL   BASE ADDRESS
LL   FLAGS
UL   BASE ADDRESS
LL   FLAGS
UL   BASE ADDRESS
LL   FLAGS

OS 1100 IBANK
PCT
IBANK
DBank
OS 1100 DBANK
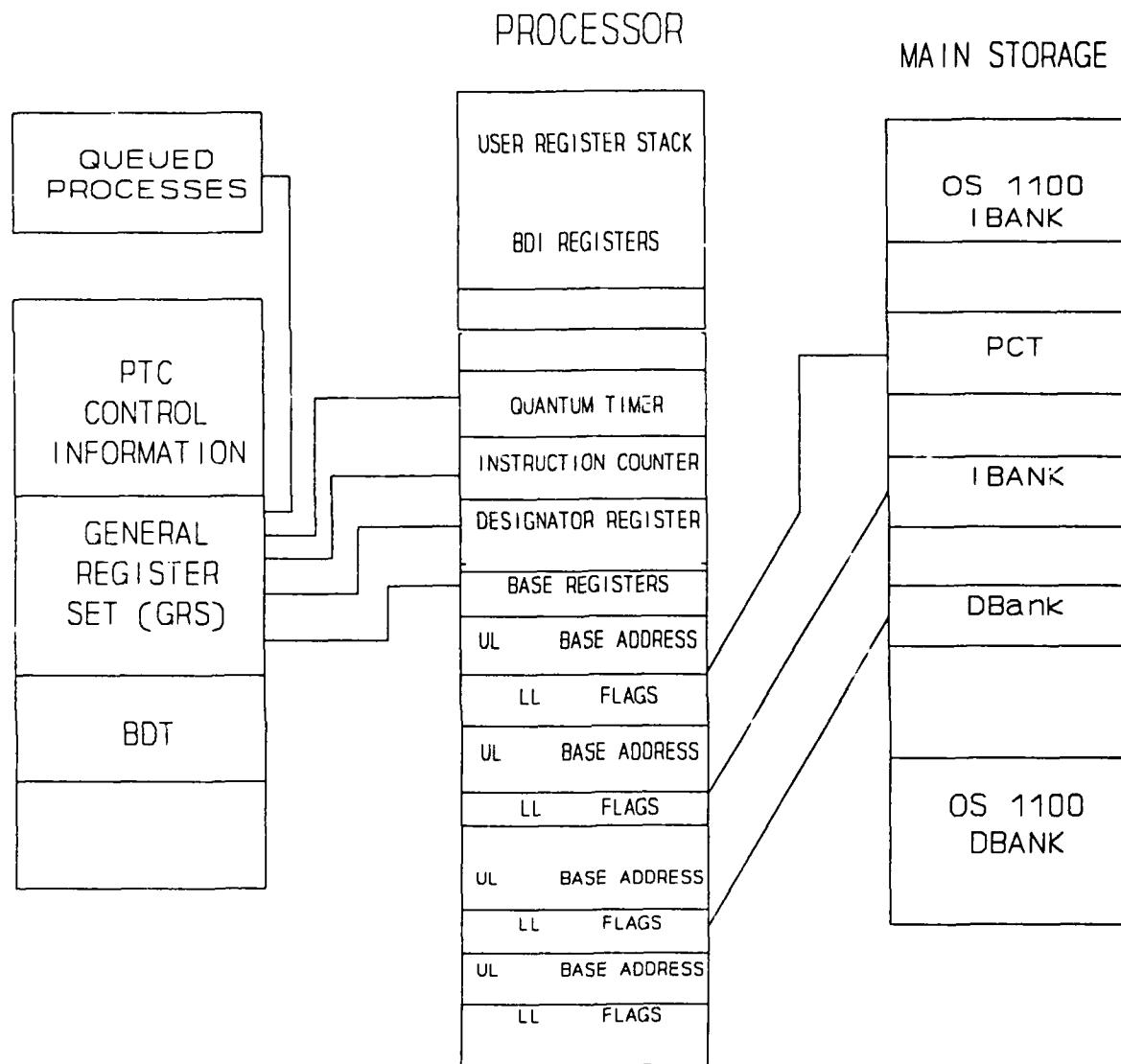
The above diagram demonstrates process activation with only 4 base registers. In extended mode 16 user base registers can exist.

Instruction Set Summary

The instruction sets of the 1100/90 and System 11 are identical as described in ASA-113 [37]. Instruction microcode is model-dependent. Differences in instruction set and

15   27 September 1989

architecture between the Series 1100 and Series 2200 are documented in ASA-114 [38]. These differences relate to the 1100/90, System 11, and 2200/200 instruction processors. The differences in instruction sets were analyzed by the Formal evaluation team and determined to be handled properly by OS 1100. For example, the 2200/200 has paging-related instructions which are not present on the Series 1100 systems. Paging capability on evaluated 2200/200 systems is not used. The model-dependent differences are not considered to be security relevant because none of the differing instructions are used to implement the security policy of the systems and none can be used to override the security policy.[1] The three models evaluated include both basic and extended mode addressing instructions, plus a group of instructions that facilitate instruction-processor control. Extended mode permits a greater range of addressability of memory banks using a larger set of base registers than is the case with basic mode.

Memory Configuration

The Series 1100[2] and 2200 systems are 36-bit word-oriented. Memory is contained in Main Storage Units (MSU). One word can contain four ASCII characters (bytes), six Fieldata (Unisys particular) characters, or a binary value.

1100/90

The 1100/90 can contain 2,097K words to 16,777K words of main storage. Memory is connected to other system components in a point-to-point manner. Memory interleaving provides eight simultaneous memory references in each MSU. Storage units interface to instruction processors, I/O processors, and System Support Processors (SSP). SSP interfaces are for partitioning and maintenance operations, which are not permitted if the system is operating in its evaluated B1 state.

2200/200

The 2200/200 can contain 512K words to 12,288K words of main storage. Memory is connected to other system components via a system bus.

System 11

The System 11 can contain 1,048K words to 4,194K words of main storage, organized in units of 1,048K-word units. Memory is connected to other system components via a system bus.

---

[1] The entire test suite developed by Unisys for the final phase of evaluation was run on all three models to verify the claim of model equivalence.

[2] 1100/90 and System 11.

Input/Output

The software handling of I/O is the same for the 1100/90, System 11, and 2200/200 (see page 33, "Interrupts"). There are differences between the models in how I/O is handled at the hardware level. Since these differences are at a level below that where the security policy is enforced, they are not security relevant [4].

Input/Output Processor (IOP)

The 1100/90, System 11, and the 2200/200 include IOPs to carry out transfers of data between main memory and the I/O subsystems. A channel program in main memory contains I/O command chains which are passed to an IOP by the IOC component of the Executive. A Universal Processor Interface (UPI) is used for communication between an IP and an IOP. Via a UPI, an IP signals an IOP to start an I/O operation.

I/O Subsystems

I/O subsystems consist of channels, control units, and devices. There is overlap among the models in their use of I/O devices.[1]

The I/O subsystems available with 1100/90 Systems include the 5057/8481 Cache/Disk System, 84XX Disk Subsystems, UNISERVO Tape Subsystems, 077X Printer Subsystems, the 0716 Card Reader Subsystem, and the DCP Systems. These subsystems interface to an I/O processor via a block multiplexer channel, a word channel, or a FIPS-compatible channel. All connections are point-to-point.

The I/O subsystems available with 2200/200 Systems include integrated disks, the 8451 Disk Subsystem, the Integrated Tape Cartridge, the UNISERVO 18 Streaming Tape Unit, the UNISERVO 22/24 Tape Subsystem attached through a Byte Peripheral Adaptor, the UNISERVO 22/24/26/28 Tape Subsystem attached through a Block Multiplexer Channel and the DCP Systems through a Byte Peripheral Adapter or a Block Multiplexer Channel. Printers and card readers are also supported.

The I/O subsystems available with System 11 include the 8451, 8463, and 8436 Disk Subsystems, a Streaming Tape with integrated control unit, the UNISERVO 22/24 or UNISERVO 22/24/26/28 Tape Subsystems, the 0789 and 0776 printers, the 0719 Card Reader Subsystem, and the DCP Systems. These subsystems interface with an I/O processor via disk controller channels, byte bus channels, or block multiplexer channels.

Communications

The Series 1100 and 2200 systems may be configured with Distributed Communications Processors (DCP) configured as front-end processors. In addition, the System 11 may

---

[1]See page A-1, "Evaluated Hardware" for a list of I/O devices in the evaluated systems.

include a Workstation Control Unit (WSCU), a Communication Line Control Unit (CLCU), or an Integrated Communications Processor (ICP). The 2200/200 may use a WSCU, but not as a communications unit, only as an interface to an onsite console. Each of these provides the communications functionality of a DCP by means of firmware emulation.

The DCP has two types of console attached to it: DCP/OS workstations (these remain inoperative while the system operates in B1 mode) and Network Management Services (NMS) consoles. These are used to control the communications configuration and are not used to control the OS 1100 system [96, 97].

Remote terminals configured on a DCP must be directly connected to the DCP. The system has not been evaluated as a network component.

Operator Console

OS 1100 supports the concept of a distinct operator's console. This is a special terminal with physical and procedural protection comparable to the TCB hardware. The 1100 Executive recognizes the console as always being available for use as an output device for system messages. It is also a place to prompt operators for guidance in carrying out certain OS 1100 responsibilities. Individual operators on OS 1100 are not required to identify or authenticate themselves to the TCB. This is in accordance with TCSEC interpretation C1-CI-04-86, which applies to B1 and below evaluated systems. This interpretation requires the maintenance of a manual log of the time and console identification corresponding to each operator's periods of console access. The Trusted Facility Manual states these requirements for OS 1100. All operator keyins are audited. Specific system requirements for operator consoles are as follows:

- 1100/90

    1. One to five keyboard/display units with optional printers in a daisy-chain (series) configuration for each operator console

    2. One to four independent operator consoles may be configured, one of which must be designated the master operator console

- 2200/200

    1. Onsite workstations hardwired to the Workstation Control Unit can be configured as a system console. A user terminal cannot be designated as a console. A console may be an SVT 1121 or UTS 20L terminal.

- System 11

    1. A UTS 20C display/keyboard with an optional printer is used as the system console

System Support Processor (SSP)

The SSP interfaces to IPs, IOPs, MSUs, and peripheral subsystems [4]. The 1100/90 can support one or two SSPs and the 2200/200 and System 11 contain one SSP on the system bus. The SSP controls maintenance and diagnostic functions, power application, mode selection (maintenance or system), partitioning, initialization, and system boot. For the evaluated system, partitioning is not permitted. The function of the SSP is to boot the system and to provide error recording and fault recovery after the system is up (in system mode). When the SSP is in maintenance mode, system controls are no longer in effect. Therefore the system is not operating in its evaluated B1 state when an SSP enters maintenance mode. The SSP has a console from which the system may be booted. This console may be used to control the OS 1100 system, by being placed into system console mode. In this mode, all OS 1100 commands are routed to the main system for processing.

Diagnostics

A hardware validation product known as DVR (Diagnostic Verification Routines) will validate correct operation of the TCB hardware and firmware. All C-Series mechanisms are tested [94]. The test procedures are manually started. The DVRs are documented by a file on the DVR release tape.

Firmware

Microcode is used to implement the OS 1100 TCB IPs and firmware is used in some of the communications and I/O subsystem controllers. Unisys was asked to supply documentation on the firmware used in OS 1100. The response was that no documentation is officially published on the firmware [94]. The position of Unisys is that the architecture-level and I/O interface-level documentation provides the design and system tests provide assurance that the I/O subsystems work properly. The low-level I/O is under control of the TCB software (see page 33, "Interrupts") and is therefore protected from unauthorized access. The evaluation team has discussed the issue of firmware documentation in the evaluation community and concludes that at the B1 level it is not practical for vendors to supply firmware documentation, and, since OS 1100 security decisions are made above the firmware level, little would be gained by examining firmware. However, the team is satisfied that firmware changes are controlled by Unisys sufficiently that integrity is ensured. A partial listing of hardware and microcode design documents has been reviewed.

Software Overview

The UNISYS OS 1100 Security Release I is actually a collection of independent products developed by separate groups within Unisys that form a single release, sometimes referred to as the "Series 1100 Operating System." In addition to a "kernel" known as the 1100 Executive, UNISYS OS 1100 Security Release I includes such products as communications software (CMS 1100 and TELCON), audit reduction tools (LA), site management software (SIMAN), and many others. See page B-1, "Evaluated Software"

for a complete list. Unisys adopted a system-level strategy for managing, integrating, testing, documenting, and releasing Series 1100 software in 1985, replacing its previous distribution technique which was based on continuous, asynchronous releases of individual components [25].

The heart of OS 1100 is the 1100 Executive which has overall control of system resources such as the IP and memory. OS 1100 is a multi-programming, multi-processing system that implements a tightly coupled symmetric multi-processing environment in hardware and software. The non-Executive components of OS 1100 which are part of the TCB support the operation and maintenance of the system in accordance with its security policy. These components are CMS 1100, COMUS, DPREP1100, FAS, IRU, UDS, MCB, PERCON, SIMAN, and TLABEL (see page 41, "Non-Executive Trusted Software").

TCB Protected Resources

Unisys has a small number of subjects and objects in the 1100 Executive; the subjects are subsystems, and the objects are files, tape volumes, and subsystems.

Definition of Subjects

The single subject type in 1100 Executive is the subsystem.

Subsystems

The term "subsystem" covers several concepts. A subsystem is a collection of banks (regions of virtual memory in the virtual address space of all users) which runs with its own protection and linking environment [15]. All the banks in a subsystem (either in executable form, or as "templates" which define the bank contents on loading) are collected into one or more files whose owner is defined to be the subsystem owner; this owner is a user-id for which there is a record in the security data base, even though there will be no user who logs onto the system using that user-id. The security attributes permitted for the subsystem are the same as those of the user-owner. The set of security attributes associated with the user-owner and thus the associated subsystem are clearance level range, compartment set, access list, and trusted privilege set. The security attributes account number and project-id are associated only with home subsystems (see next paragraph).

Subsystems are categorized as follows: home, common bank, and library. Processes may execute only within home and common bank subsystems. Thus, only home and common bank subsystems are subjects. Library subsystems are containers for public data; hence no process state changes are necessary.

HOME SUBSYSTEM

When a batch or demand run or TIP session is opened (also, for TIP, every transaction initiation) the Executive establishes the home subsystem with a unique name. This home subsystem includes an initial protection environment with a collection of control tables in

memory that define the user's security attributes: clearance level, compartment set, and privileges. The files to which the user's run or TIP session has access are also defined in control tables.

The security attributes of the home subsystem are derived from the user's security profile. Included in the static security attributes is the privilege set. The home subsystem's privilege set may include privileges belonging to the trusted privilege set, thus making the home subsystem a trusted subsystem. If the home subsystem's privilege set does not include privileges belonging to the trusted privilege set, then the home subsystem is untrusted. The executing attributes of the home subsystem are those specified during the log-on process.

The home subsystem is always in one of two states: idle or acting on behalf of the user. When the home subsystem is idle, no processes are within the subsystem. When the home subsystem is acting on behalf of the user, one or more processes are within the home subsystem. In batch or demand home subsystems, processes are initiated by ECL requests or during task execution. In TIP home subsystems, processes are initiated as a result of transaction execution. Within a user's home subsystem is the Program Control Table (PCT) for the session or TIP transaction. Part of the PCT indicates what files the user has currently assigned.

COMMON BANK SUBSYSTEMS

All processes execute within the home subsystem until a bank protected by a common bank subsystem is referenced. The common bank subsystem is always in one of three states: defined, activated but idle, or activated and acting on behalf of a user.

A common bank subsystem is defined when its security attributes are defined and the common bank or banks are installed (see page 43, "Computerized Onsite Maintenance for User Systems (COMUS)") in a file or files owned by the user-id specified in the security attributes for the common bank subsystem. A subsystem may access files identified in a calling home subsystem's PCT or it may assign its own files using the ER TRON$.

When access to the common bank subsystem is made for the first time after system initialization, the Executive activates the common bank subsystem (it is activated but idle). When the common bank subsystem has been activated and is idle, no processes are within the subsystem. As part of the common bank subsystem activation, the security environment for the common bank subsystem is established. The security attributes of the common bank subsystem are derived from the subsystem record in SACRD$ (see page 29, "Security Database") and the accessing home subsystem. The subsystem record is the user-id record of the owner of the file or files which contain the common bank or banks. The executing attributes of the common bank subsystem are established as follows:

1. The executing project-id and account number are taken from the home subsystem.

2. The executing privilege set and user-id are the privilege set and user-id in the subsystem record.

3. The executing clearance level and compartment set are the maximum clearance level and compartment set in the subsystem record.

A common bank subsystem is activated and acting on the behalf of a user(s) when a process(s) is within the subsystem. Processes within common bank subsystems are always user activities resulting from task or transaction executions that were initiated in one or more home subsystems. Before a process is allowed entry into a common bank subsystem, validation occurs between the calling subsystem's security attributes and the common bank subsystem's security attributes.

When a common bank subsystem is defined, its entry is protected by a gate bank (see page 9, "Instruction Processor (IP) and Main Storage") which has a lock (see page 10, "Use of Keys and Locks") protecting it. An activity attempting to enter the subsystem must have a key that will fit the gate bank's lock. The gate bank has another key stored within it which is used once the subsystem has been entered. All banks within a given subsystem are protected by locks which have the same domain number. This will permit a key-to-lock match for any activity executing within the subsystem. By separating common banks into separately-locked subsystems, OS 1100 is able to provide controlled access to code and data shared between subjects. The installation of common-bank subsystems, and the assignment of security attributes to them, is a function that requires actions by the SSA [17, 15, and 49].

PROCESSES

Subsystems are named entities acting on behalf of users, thus making them surrogate users. Each action performed by a subsystem on behalf of a user is a process. A process is initiated by a task execution, a transaction execution, or an ECL statement execution. Thus, at any point in time, a subsystem may have zero or more processes within it. A process executing in a common-bank subsystem has the ability to switch between either of two file control tables, its own and that of its caller. When a process (whether user activity, transaction request, or ECL statement) executes within a subsystem, it acts with the subsystem's security attributes.

The set of objects that a subsystem can access is defined by the subsystem's security attributes and the address space of the processes within it. Each subsystem has an address space which limits the main storage, assigned files, and tape volumes that can be accessed by processes within the subsystem. Each process within a subsystem has an address space that equals or is contained within the subsystem's address space. The address space of the process consists of the memory addresses it is authorized to access, and the files and tape volumes that it has already been validated to access. The memory addresses that are authorized are determined by the subsystem, which may change when a user process transfers from one subsystem to another. Information as to which files and tape volumes can be accessed (those which have already been assigned), is contained collectively in the PCT of the home subsystem and in the name section of a common bank subsystem.

SUBSYSTEM TRANSITION

Invocation of a subsystem involves two stages: "activation" of the subsystem (which entails loading its banks and establishing its initial security attributes) and "transition". "Activation" if successful, is always followed by "transition" into the subsystem (which entails transferring control from the calling subsystem to the called one).

When a process executing in a subsystem attempts to transfer control to a different subsystem, mandatory access checks are made and, if successful, validation occurs to determine whether or not the referencing subsystem has enter access (see page 10, "Use of Keys and Locks") to the referenced subsystem. If all access is permitted (see page 58, "Access to Subsystems"), the subsystem is activated (instantiated) and assigned an executing security level that is determined by whether the (new) subsystem is trusted or not:

- if the subsystem is an untrusted or a single-level trusted subsystem, it is assigned the same security level and privilege set as that of the calling subsystem

- if the subsystem is a multilevel trusted subsystem, it is assigned the highest security level (which includes the maximum compartment set) and the maximum privilege set of its owner

When a transition is made from the home subsystem to a trusted subsystem (see below), the trusted subsystem has access to the files described in the PCT. Also, an Executive activity doing work on the user's behalf may attach itself to the user's PCT and execute with the user's security characteristics. This limits the security domain to the appropriate boundaries for the user.

Trusted Subsystems

A trusted subsystem is one which is allowed to perform some security-relevant or protection-critical function. Such a subsystem operates with user Processor Privilege (see page 9, "Instruction Processor (IP) and Main Storage"), and has no inherent access rights to objects different than the rights of ordinary subsystems. However, these subsystems are given privileges to bypass the security constraints as needed to perform their functions. These privileges are taken from the Trusted Privilege Set (the TPS), which consists of the following privileges (see also page 63, "Privileges"):

    a.   SSCCL - Bypass clearance level validation (within range) and change
            run clearance level
    b.   SSBAFC - Bypass access list evaluation
    c.   SSBYCL - Bypass clearance level validation (outside range)
    d.   SSBYCOMP - Bypass compartment validation
    e.   SSADID - Absolute device-id assignment
    f.   SSBVOLCHK - Bypass volume label validation
    g.   SSBYPASSOWNR - Bypass owner validation

      h.   SSSMOQUE - Allows print file queue entries to be accessed or deleted, regardless of security level

      i.   SSDBACK - Allows FAS backup tape number change in MFD

These trusted privileges are grouped into two classes: (A) the privileges that override the policy rules directly (SSCCL, SSBYCOMP, SSBYPASSOWNR, and SSBAFC), and (B) those that act only through Executive Requests and control statements.[1] There is one additional privilege, SSSSCALLANY, which completely bypasses all subsystem transition validations. The other TPS privileges only bypass some of the security validations. The TFM states that the risk of misusing SSSSCALLANY is to declassify data. It is clearly security critical.[2]

Two types of trusted subsystems are defined: single-level and multilevel access trusted subsystems. These two types of subsystem have been defined to increase the granularity of security domains within the TCB for enforcing the least-privilege concept. They control which types of subsystem may access them (single- or multilevel access, trusted or untrusted). A single-level access trusted subsystem (SLSS) is one whose privilege set contains some proper subset of the policy override privilege set (A), and any other members (B) of the Trusted Privilege Set. Only a calling subsystem whose security level equals the security level of a SLSS can gain access. A multilevel access trusted subsystem (MLSS) is one whose privilege set contains ALL of the policy override privilege set (A), and any other members (B) of the Trusted Privilege Set. Callers whose executing security level and security level ranges lie within that of the MLSS gain access to such a subsystem. All other subsystems (i.e., those with no privileges from the Trusted Privilege Set) are untrusted.

The transition validation rules for entry into a trusted subsystem are as follows:

- Untrusted to untrusted or to SLSS: Executing clearance and compartment sets must be equal

- Untrusted to MLSS: Executing clearance and compartment set of the caller must be within the range of the MLSS

---

[1] The privilege SSBYCL, which has been used in the past as a means to bypass the clearance level validation, permits access outside a subject's clearance range. Unisys has been discouraged from using it and encouraged to use the privilege SSCCL instead. The TFM recommends using correct clearance level ranges for subsystems to aviod the need for SSBYCL. The TCB components use SSCCL, although several of them are also given SSBYCL since the components need the Trusted Privilege Set, of which SSBYCL is a member.

[2] Only the Integrated Recovery Utility (see page 44, "Integrated Recovery Utility (IRU)") uses SSSSCALLANY, which it needs to recover databases which may be at different security levlels.

- SLSS to untrusted: Not allowed

- SLSS to SLSS: Executing clearance and compartment sets must be equal, and the caller's privileges from the TPS must be a subset of those of the called subsystem

- SLSS to MLSS: Executing clearance and compartment set of the caller must be within the range of the MLSS, and the caller's TPS must be a subset of the MLSS TPS

- MLSS to untrusted: Not allowed

- MLSS to SLSS: Not allowed

- MLSS to MLSS: Caller's executing clearance and compartment set, as well as TPS, must be a subset of the called MLSS's sets.

Definition of Objects

Unisys has identified three types of protected objects in OS 1100: files, tape volumes, and subsystems.[1] References to each type of object are mediated by the 1100 Executive, with mandatory and discretionary rules applied. For tape volumes and files, a newly created object inherits the executing clearance level and compartment set of its creator. For subsystems, the executing clearance level and compartment set are established at subsystem activation, as discussed below. In all cases, discretionary access information must be either explicitly set on creation, or access will default to private, i.e., accessible only to the object's creator. Specifics are discussed below.

The security attributes associated with an object are stored in a security record in the security database. The specifics of this database are discussed on page 29, "Security Database". Records in the security database are controlled by the 1100 Executive and can only be modified under certain conditions, which vary depending on the object type with which the record is associated. Object security records are always owned by the owner of the object with which that record is associated. User security records, i.e., those that define the security characteristics that a user may assume at logon or a subsystem assumes at activation, are owned by the SSA or by a sub-SSA.

A security record is created or deleted by the 1100 Executive when the object it represents is created (or first written to in the case of a tape volume) or deleted, respectively. Note that to ensure unique ACR (see page 56, "Discretionary Access Control") names and user names throughout the lifetime of the system, deleted user and ACR records are marked as inaccessible but are not actually purged from the database.

---

[1]Demountable disk packs are not defined as protected objects, since the files stored on them are protected objects. Files stored on tape reels are not defined as protected objects and thus tape volumes are defined as protected objects.

Files

A file is an organized collection of data, treated as a unit and stored in such a manner as to facilitate the retrieval of each individual data item [4]. OS 1100 supports two basic types of files, data files and program files. The system makes use of the distinction between data and program files in that program files can provide the user with added convenience in data management. For example, language compilers use program files to contain symbolic, relocatable, and absolute elements. The user need specify only one name for an element and the system will keep track of different elements with the same name which have different content types. A program file is typically used to store one or more modules of program code in symbolic, relocatable, or absolute form.[1] Program file elements are referenced by a compound name which consists of the file name and an element name. Security attributes are associated with the complete program file, not the individual elements contained within the file [16].

Data files typically contain binary data or text and are referenced by a simple (not compound) file name. Mandatory and discretionary security attributes are associated with the data file.

Data files and program files can be stored on disk (also called mass storage) or tape. However, files stored on tape are protected at the tape volume level (unless the file is copied to a disk file).

In addition to program and data files, the 1100 Executive also permits exclusive assignment (see page 58, "Access to Files") of certain files to the TIP transaction processing environment. These files contain data internally divided into units known as TIP files. All TIP files contained within one Executive file have homogeneous security attributes, which are actually associated with the enveloping Executive file.

Files are created with ECL (see page 35, "Executive Interfaces") statements and may be either temporary or permanent (cataloged). Temporary files exist only for the duration of a user's run, and are available only to the creator. A security level and an access list are associated with permanent files.[2] The security attributes associated with files are: clearance level, compartment set, and access list. The file's clearance level and compartment set come from the executing clearance level and compartment set of the subject creating the

---

[1]Symbolic elements contain text and relocatable or absolute elements contain binary information. The three types of element may be mixed within a program file.

[2]Temporay files have the mandatory security attributes of the user's home subsystem if assigned by an activity in the home subsystem. This MAC information is kept in the PCT for the run. Temporary files assigned by a common-bank subsystem have the mandatory security attributes of the common-bank subsystem. The DAC on all of the subsystem's temporary files is private to the subsystem. If a common-bank subsystem transitions to another common-bank subsystem, the new common-bank subsystem can ahve MAC and DAC access to the temporary files of he old common-bank subsystem.

file, and may only be changed by someone with security database administrator privilege whose current security level equals the current level of the file. The distribution of this privilege is controlled by the SSA, and any user who has this privilege is considered to be an SSA. When a subject with security database administrator privilege (an SSA) modifies the mandatory attributes associated with a file, the new clearance level of the file must fall within the range of permissible clearance levels for the SSA, and the new compartment set must be a subset of the maximum compartment set allowed for this SSA. Such reclassification actions are audited [4].

A file's discretionary attributes may be modified by the owner of the file, or by an SSA. The clearance level and compartment set of the subject attempting the modification must equal the clearance level and compartment set of the file.

The owner of a file may pass ownership to another user if the original owner's current security attributes dominate the new owner's maximum, which in turn must dominate those associated with the file. An ACR attached to a file will be removed if the owner changes since the owner of a file and the owner of that file's ACR must be the same; in this case, the file becomes private [16].

Details of the access types and rules are described elsewhere in this report (see page 58, "Access to Files").

Tape Volumes

A tape volume is a single physical reel of tape. A tape volume is identified by a reel number recorded in the tape header. A tape volume is sometimes called a tape file, since a tape volume is referenced with the syntactic equivalent of a file name in ECL statements. Tape volumes are not exactly the same as tape files, however, because a set of related tape volumes may be cataloged under a single tape file name in the MFD (Master File directory). If a tape file is cataloged, its MFD entry includes one or more reel numbers which correspond to tape volumes.

The security attributes associated with a tape volume are clearance level, compartment set and access list. These attributes, in addition to other attributes such as owner name and reel number, are always recorded in the tape header. If the tape is cataloged (i.e., not temporary), all attributes are also recorded in the MFD [42, 77]. The main difference between the MFD entry for a disk (mass storage) file and that of a tape volume, is that the tape volume appears as a reel number within an MFD entry for a tape file. Tape files may include several volumes. This permits individual files on tape to span volumes. However, each volume is individually protected by OS 1100 access control.

MAC and DAC may be initialized for a tape volume either before the tape is first accessed for use (prelabelling) or when the tape is first accessed for use. It is only possible to change MAC or DAC on a tape volume that is new or that is to be completely rewritten. The recommended method to change an existing tape's MAC or DAC is to copy the files on a tape volume to another volume, change the label and possibly the ACR, and copy the files back onto the original tape volume.

The association of security attributes with a tape volume occurs in one of two ways. In the first method, at the first write to the tape, the TCB will set the mandatory attributes of the tape to those of the subject responsible for the write operation. At the first write, the discretionary control on a tape volume is set to private. DAC may later be changed using the TLABEL utility [42, 77] but, if the user does so, existing data on the tape is lost (because the label must be rewritten on the tape, which causes all following data to be inaccessible).

Alternatively, tapes may be prelabeled. A prelabeled tape may contain either an ISO VOL1 header, specifying only an owner for the tape but no security attributes, or both VOL1 and VOL3 headers which specifies owner and a security level; it may alternatively be given a security label, but with no owner identified. If DAC is specified in a tape label, it defaults to private. Any such labeling operation requires privileges controlled by the SSA.

If only VOL1 is written to a tape, the security label (VOL3) will be added at the first write, with the tape's clearance level and compartment set taken from the creator's executing clearance level and compartment set, and the volume is marked private. If VOL1 and VOL3 are written, the discretionary controls will be set to private [42]. In any case, the user can use the TLABEL command to set the discretionary controls as he wishes.

Tape volume mandatory attributes can be modified only by an SSA, and then only if the SSA's current security level equals the tape volume's level. The new clearance level of the tape volume must fall within the SSA's defined range of levels. Likewise, the new compartment set for the tape volume must be a subset of the SSA's maximum compartment set [4].

As with files, the discretionary attributes of a tape volume may be modified by the owner of the tape volume, or by an SSA. The clearance level and compartment set of the owner or SSA attempting the modification must equal the clearance level and compartment set of the tape volume.

All data is inaccessible on the tape if MAC or DAC is changed.

Details of the access types and rules are described elsewhere in this report (see page 59, "Access to Tape Volumes").

Subsystems

A subsystem is the only entity in OS 1100 that acts as both a subject and an object. The subsystem is defined as an object because its banks may contain writable data protected by the security policy and because one subsystem can access another. A general discussion of subsystems can be found on page 20, "Subsystems". Recall from that discussion that a subsystem's security attributes, when a subsystem is treated as a subject,

are the security attributes[1] associated with a home subsystem and with the owner of the file that contains the common banks for the subsystem [4]. When treated as an object, only the static security attributes[2] associated with the owner are used.

All subsystems are installed by the SSA. The security record for the user-id which is to be the owner of the subsystem must already exist. This user-id is not generally used as the name of a home subsystem because no user logs on with this user-id.

Subsystems will be installed by the SSA as trusted subsystems if they are given privileges from the Trusted Privilege Set. See page 23, "Trusted Subsystems" for more information on this topic. The SSA may not install a new trusted subsystem into the TCB, in addition to those evaluated, and maintain his particular system's B1 rating. The mandatory security attributes, privileges, project-id, and account number associated with a subsystem may only be modified by an SSA. In addition, that SSA's security attributes must dominate the new values of any attributes being modified, the subsystem's new privilege set must be a subset of the modifying SSA's privilege set, and the new project-id or account number must be valid for the SSA. As with files and tape volumes, the discretionary attributes of a subsystem may be modified by an SSA. In any case, the clearance level and compartment set of the user-id attempting the modification must equal the clearance level and compartment set of the subsystem.

Details of the access types and rules are described elsewhere in this report (see page 58, "Access to Subsystems").

Security Attributes

This section describes how security attributes are associated with the subjects and objects in OS 1100.

Security Database

For convenience, the vendor and the evaluation team use the term "security database" to refer to the collection of security-relevant information associated with subjects and objects in OS 1100. In actuality, no single security database exists. Instead, security-relevant information is stored in different locations depending on the type of entity with which that information is associated. This section discusses the various security-relevant control information. For more information on how access to the security database is restricted, see page 53, "Access Control".

---

[1] project-id, account number, clearance level range, executing clearance level, compartment set, executing compartment set, security record owner, and privilege set.

[2] clearance level, compartment set, and privilege set.

The security information contained in the files and records which make up the security database is as follows:

- SYS$*TSS$FILE (a file) contains user logon attributes. These attributes are user-ids, passwords, account numbers, project-ids, password expirations, application control information, console capability, and minimum password length. Note that this table must remain consistent with the information in SYS$*SACRD$. This is achieved by saving and restoring them together.

- SYS$*SACRD$ (a file) contains user security attributes and Access Control Records (ACRs). The user security attributes are clearance level range, maximum compartment set, subsystem information, number of ACRs, and privileges.

- SYS$*SMDTF$ (a file) contains the system's definition of symbolic clearance levels and the compartment set. Note that this table must remain consistent with the information in SYS$$*SACRD$ so that a user's security attributes reflect the correct compartment definitions. This is achieved by saving and restoring them together.

- SYS$*MFDF$$ (a file) is the Master File Directory, which contains file security attributes (classification level, compartment set and discretionary access control).

- SYS$*ACCOUNT$R1 (a file) is the Summary Account File, which relates account numbers to associated user-ids, and controls a special privilege.[1]

Additional security information is contained in the System Log File, which contains the system audit trail information, and in tape labels, which contain file security attributes similar to those in the MFD.

In addition to these external records, security information is placed in control structures created by the system during its normal execution, which are sometimes referred to as security records.

Security Records for Objects

The security record for a tape volume is stored in the tape volume's header, and, for cataloged tapes, the MFD as well. The tape volumes appear as reel numbers in the MFD entry of a tape file. If an ER (see page 35, "Executive Interfaces") is used to modify the information in the MFD without also changing the information on a cataloged tape volume, the 1100 Executive will detect the inconsistency on the next attempt to access that tape using the MFD, and reject the access.

---

[1]execution in real-time mode, with its access to shared memory locations.

The security record for a cataloged disk file is stored in the MFD. Security information for cataloged and temporary disk files is derived from the security attributes of the creating subsystem.

Information making up the security records associated with a subsystem, in its incarnation as an object, is stored in the security record associated with the files encapsulating the subsystem (i.e., the MFD entries).

TIP terminals are not defined as named objects because they cannot be written to by the unprivileged user.

Security Records for Subjects

The only subject in the system is the subsystem. During the logon step identification and authentication occurs, which utilizes the user security information to validate a user's request for access to the system (see page 53, "Access Control") before any subject exists. This identification and authentication step also validates the user's executing clearance level and compartment set. After logon, a home subsystem is activated for the user.

The security information associated with subsystems, acting as subjects, is a combination of information associated with the subsystem and that associated with the home subsystem of the process which is executing. The portion obtained from the home subsystem is the project-id and account number. The remaining information is taken from the subsystem, which derives its security record from that of the owner of the subsystem. This includes the executing clearance level and clearance level range, the executing compartment set, the name (for use in discretionary access decisions), and the privilege set.

TCB Software

The Trusted Computing Base for OS 1100 is by the nature of the system large, multi-threaded and complex. The software included within the TCB supports a number of tasks and utilizes global variables within the major modules of the TCB. The services provided and a short reason for the inclusion of various programs in the TCB are given below.

The 1100 Executive comprises the major part of the TCB. As the control program for OS 1100 it makes all security decisions along with all resource and operation decisions. The 1100 Executive is coded in a macro-assembler language and in PLUS (a PL/1 like higher level language defined and implemented by Unisys). The interfaces and programs implemented reflect the history of this system and its evolution from 1964 to the present.

Another element within the TCB is the software to support the System Support Processor (SSP). The SSP is used in booting the system. While the SSP is booting the system, performing recovery, or running diagnostics, it has absolute control over the hardware. Many of the diagnostics and system test programs are run by the operator and site engineers using the SSP. In diagnostic mode, the operator using the SSP is allowed to do memory reads and writes to absolute addresses and therefore must be trusted to prevent

loss or compromise of data. This also requires that the SSP and its software be included in the TCB.

Executive Software

OS 1100 implements a symmetric, tightly-coupled, multi-processor configuration in hardware and software. The system runs all processors with a single copy of the 1100 Executive in shared memory. Through the use of test-and-set instructions and system coding conventions, the 1100 Executive ensures that critical paths are single thread. In addition, data structures referenced by interrupt pre-processing (see page 33, "Interrupts") are created on a per-processor basis. Processors are prevented from interfering with each other via hardware interfaces. The 1100 Executive maintains a single memory-resident clock which is monotonically increasing.

The structuring of the 1100 Executive is such that all Executive activities (processes) run with attributes that allow access to all of the system. Thus while some parts of the 1100 Executive are not security relevant, they are part of the TCB. The Executive runs with all privileges (except SSBVOLCHK - bypass tape volume check), access to all secured ERs, ring=0, domain=1, processor privilege=0 (Executive mode) and security attributes of security-privileged system high (see page D-1, "Glossary of Terms"). The Executive does not have the SSBVOLCHK privilege so that all tape assignments from within the Executive will perform a reel number validation. This operation ensures against an operator error.

All 1100 Executive data space is restricted from user access through the use of the ring (=0) and domain (=1) (page 9, "Instruction Processor (IP) and Main Storage"). The exception to this is the Program Control Table (PCT), the control structure for a run. Since it maintains the state of a run, it is part of the address space for the run and is readable only by that run (resides at level 2 of the address tree (see page 8, "C-Series Addressing") and thus only is part of that run's address space).

The Executive is structured into a set of components. These components are made up of independently compiled/assembled entities called elements. All of these elements are collected (linked) together into a single executable entity that is loaded at system boot time (initial program load).

Within the Executive is a central security complex [4].[1] This complex contains the security validation routines. A unique interface, in the form of the SUVAL$ ER (see page 35, "Executive Interfaces"), is provided for security validation by programs outside of the TCB.[2] Within the Executive, but outside the central security complex, either the LINK entrance to the SUVAL$ validation routines or the resident Executive element SSRVCR

---

[1] Executive components which lie outside the security complex include the file control, scheduling, and symbiont complexes.

[2] The TCB does not make use of ER SUVAL$.

may be used. ER SUVAL$, as well as its alternate Executive element, performs access validation, ER validation, and privilege validation. Central security complex validation includes the ER SUVAL$ validations plus additional ER and privilege checks, tape assignment and file access, MAC, DAC, subsystem transition, and security database locking.

The Security Component of the Executive provides interfaces for maintaining the security database, controls user access to the system, validates access to objects, and logs security-relevant events.

Interrupts

The Executive can be viewed as being broken into three logical levels of execution. OS 1100 is an interrupt-driven system and these three levels are interrupt pre-processing, interrupt post-processing, and Executive worker activities. Interrupt pre-processing is the level that gets control from the hardware. This processing is normally very short and consists only of the work necessary to capture pertinent interrupt-related information and place it into software queues for later processing. This code is executed with interrupts disabled to ensure single-thread operation. No program state save is required since this processing uses only a predefined subset (established by convention) of the Executive register set. While this execution effectively runs with the Executive attributes, no accesses to objects outside of the TCB are made that require security validation. Interrupt pre-processing runs in extended mode (page 8, "C-Series Addressing").

Interrupt post-processing is accomplished by Executive processes that are logically non-interruptible (except by interrupt pre-processing). This processing is done with the Executive (not user) attributes and at a high priority. Post-processing performs some of the security validations required for user requests and performs the physical I/O to files as requested by users. Post-processing also performs short-duration tasks, but not so short as required for interrupt pre-processing. Post-processing is primarily oriented towards ensuring maximum I/O activity and doing the work necessary to initiate Executive worker activities to do the more lengthy or interruptible tasks.[1] Post-processing activities share the Executive register set, but use a mutually exclusive subset from interrupt pre-processing. User context is maintained in the user registers.

Interrupts can be categorized into groups, primarily based on the 1100 Executive components that handle the post-processing of those interrupts.

1. Hardware - These interrupts, indicating hardware errors, are processed by the Hardware Fault Control complex of the 1100 Executive. Each hardware error generates a system log entry. Where possible, the 1100 Executive attempts to recover from the error, including dynamically isolating a component or memory module. This may require providing

---

[1]Executive worker activities are those run on behalf of a user. They use the user register set and have the user's security characteristics when accessing objects on behalf of the user.

affected users with a contingency.[1] Diagnostic routines may be initiated to provide information for problem resolution. Where recovery or isolation of the problem is not possible, the 1100 Executive will halt the system.

2. External (I/O) - These interrupts are generated from the channels and are generally handled by the I/O Control complex.[2] This complex insures that the I/O channel activity is maximized, by initiating an I/O operation as soon as the previous operation on that channel is complete. In the case of interrupts that indicate error conditions, the I/O complex will attempt to recover from those errors by retrying the operation. When these retries do not resolve the situation, the requesting user is notified via an error status on the I/O request. If necessary, areas of a disk may be marked as unusable or devices may be dynamically removed from the system, to allow the system to continue operation. The system stops when an error affects the integrity of system data structures.

3. Software - These interrupts provide Executive interface- and program-control functions and also indicate program faults. Included in this category are the timer interrupts used by the Executive to timeslice users and to compute resource usage. The program control interrupts include the interface to the Executive (SIGNAL interrupt), synchronization (test and set interrupt), and various stack overflow/underflow conditions.

The program-fault interrupts include errors in the use of certain types of instructions (floating point or field instructions) as well as the security-related interrupts of invalid instruction, reference violation, or addressing exception. The Executive identifies the activity associated with the particular interrupt and provides status information to the user through the error-reporting mechanism. In severe cases, the activity may be given an abort condition that leads to program termination, and run termination for batch runs. When the Executive receives a program fault while an Executive process is active, a log entry is generated and the system is stopped.

The interrupt mechanism is instrumental in the process isolation and enforcement of the security policy that the 1100 Executive carries out. The Executive establishes the execution environment for a program, including setting up base registers to indicate the instruction and data areas and the allowed accesses to these areas. Accesses to areas are controlled by the hardware, such that attempts to access areas outside of those specifically identified, or to improperly reference an area (e.g., attempt a write to a write-protected area) will result in an interrupt and appropriate action by the Executive.

---

[1] i.e., a unit of code to handle error recovery

[2] The security-relevant aspects of I/O reside in the Executive component Logical I/O (IOL). This component makes all necessary security checks before calling I/O Control (IOC) to initiate an I/O operation. IOC handles device-independent I/O control. A channel program is either passed to IOC or it calls I/O Peripherals (IOP, not to be confused with the I/O Processor hardware) to build a channel program. Users cannot provide their own channel programs.

Execution and Subsystems

All activities within the system are associated with a subsystem (see page 20, "Subsystems"). Associated with a subsystem are a user-id and a set of security attributes. A logical subsystem associated with the 1100 Executive acts as the controlling entity for all 1100 Executive activities that have the Executive attributes. Each activity, whether Executive or user, runs with the attributes of the subsystem in which its code resides.

A user run is a list of tasks executed sequentially from beginning to end. Each task is executed by at least one activity (programs may have multiple activities), which begins execution in the home subsystem and which transitions to other subsystems as needed. User tasks are kept separate from each other via separate data structures and address spaces.

The Executive component that implements and maintains subsystem control structures is the Subsystem Manager (SSM). The SSM creates an external gate on the first reference to a common bank. External gates remain memory resident until system boot. The SSM has no interfaces external to the Executive. The SSM provides service routines to build subsystem structures and to retrieve information pertaining to subsystems and the SSM manages the subsystem transition process in Executive code called the Intercept Routine. When a user process references a common bank, hardware traps the request to the Intercept Routine. If the common bank subsystem has not been referenced previously by any user, the gate manager is called which builds the gate bank from values established when the common bank subsystem was installed (see page 43, "Computerized Onsite Maintenance for User Systems (COMUS)"). If the gate exists, the Intercept Routine looks into the acceleration buffer in the calling routine's Program Control Table (PCT) to determine whether or not the calling subsystem previously has been granted access. If so, the Intercept Routine grants access immediately. If access has not been granted to the common-bank subsystem, the Intercept Routine uses SUVAL$ routines which are common to security validation for all non-Executive subsystems. If the SUVAL$ routines grant access, the Intercept Routine places the common-bank subsystem identifier into the acceleration buffer of the caller's PCT and immediately grants access. Also, a Subsystem Security log entry is made indicating the results of validation on the initial access request.

Executive activities operate at security-privileged system high. Since the user-id "Exec" is cleared to clearance level 63 (the highest) and ALL compartments, Executive activities have access to all data in the system.

Executive Interfaces

The design document [4] describes three separate means of interfacing with the TCB. These are a program callable interface (the Executive Request (ER) mechanism), a control language interface (Executive Control Language (ECL)) and operator-console keyins.

Other interfaces are $$ commands, Executive interrupts, MCB $Action commands, LBJ instructions, log-on, and TIP file execute access (33). Each of these mechanisms will be described in the following sections.[1]

EXECUTIVE REQUESTS (ER)

The security relevance of ERs is that they provide a controlled interface to the Executive itself or to the central security complex of the Executive (from other parts of the Executive).

An ER is an instruction that causes an interrupt to the Executive. The address field of this instruction is an index indicating which Executive service is being requested. The Executive maintains a table of all defined ERs and the address of the code module that will process each request. The ER interrupt post-processing routine will validate the index and, based on information from the table, determine what processing must be done.

The ER Programmer Reference Manual [81] describes three different categories of ERs. Immediate ERs are completely processed in interrupt post-processing with control being returned immediately to the requester. The requester is not deactivated and there is no need for the requester's state to be saved, as the execution is simply deferred to a higher-priority process executing with the Executive's attributes and using the Executive register set. Asynchronous ERs allow the requesting activity to continue execution while the requested service is being performed. In this case, once the initial post-processing is done and the requested service is initiated (Executive worker activity initiated) control is returned to the requesting user. Synchronous ERs cause the requesting user to be deactivated, the requesting activity state to be saved, and the initiation of an Executive worker activity to process the request.

The ER interrupt is captured and processed by an interrupt post-processing routine in the Executive. This routine will:

- generally perform parameter packet[2] address checks (some specific ERs do not use a packet and others may choose to handle the address checks locally)

- handle the deactivation of the requesting user process if necessary

- initiate the 1100 Executive worker process that will handle the request, if there is one (some simple requests are handled immediately and control returned to the user)

---

[1]Components of the TCB which are outside the Executive use the ER mechanism to access the Executive or they share memory and files with the Executive.

[2]The set of parameters passed to the ER code in the Executive.

- perform the validation required for a secured ER. Refer to page 63, "Privileges" for a discussion of secured ERs.

While each ER is different in its input requirements and subsequent parameter checking, for those ERs that request parameter validation, the post-processing routine will validate that the packet does exist in space owned by the calling user and that the required access (read and/or write) to the packet is allowed. Executive accesses to the user packet use absolute addressing mode. The Executive validates all accesses against the user address space and then converts them to absolute addresses. The Executive will not write to a write-protected bank. In addition, the common packet checks[1] allow the Executive to detect user address errors in a more efficient manner. Verification of individual parameters within the parameter packet is left to the 1100 Executive component responsible for processing the individual ER.

If necessary, the 1100 Executive component processing a request acquires data space from the Executive buffer pool and transfers the parameter packet into that data space. All parameter validations are done against this copy of the parameter packet. Upon completion of the request, the ER processing code places any status or return information into the user data space. In this manner, the 1100 Executive avoids the TOCTTOU (time-of-check-to-time-of-use) problem. In those cases where parameters are passed in registers, the parameter validation is either done immediately before control is given to the Executive worker activity acting on behalf of the user, or the calling activity is deactivated and the registers are saved in the 1100 Executive-owned data structures. In the latter case any parameter value validations are done against these Executive-owned data structures.

In addition to the above actions, the ER-definition table defines whether an ER is a controlled (secured) ER. If the ER is a controlled ER, the initiation software will call a security access module to determine if the user has access to the requested controlled ER. If not, a security-abort condition occurs. The security-abort condition indicates that program execution is to be terminated immediately. All activities of the executing program are terminated. In the case of a batch run, the run is also terminated. All security-abort conditions are audited.

Access to special functions is controlled by the code for each ER. All ER code runs in Executive activities (worker activities) operating on behalf of a user.

CONTROL STATEMENTS

Several forms of control statement are available to the user. Network[2] control statements are handled external to the Executive in the front-end processor (DCP or equivalent, see

---

[1]The checks that are applicable to all packets, e.g., maximum size of fields.

[2]The term network as used here refers only to directly connected terminals. OS 1100 was not evaluated as a network component.

page 41, "Communications Management System (CMS 1100)"). These commands are distinguished by the use of $$ as a command prefix. A second form of control statement is the transparent control statement. These commands are distinguished by the use of the @@ prefix to the command and are handled by CMS 1100 or the Remote Symbiont Interface (RSI) component of the Executive. The Executive Control Language (ECL), which is the final form of control statement, is the job control language. These commands are distinguished by the @ prefix. Each of these types of commands will be discussed in more detail in the following paragraphs.

Network control statements are received and processed by the TELCON software which runs in the front-end processor. These commands are used for network signon and signoff, opening and closing TELCON sessions, and UTS terminal emulation.

The RSI commands are available to a demand run and in a limited form (@@TOUT) to a TIP session. These commands are used to control interaction with the terminal, performing such functions as terminal text control and auxiliary device control. They also allow a subset of the ECL commands to be executed in parallel with other run processing. The design document [4] discusses the security-relevant aspects of these commands.

ECL commands are processed by the Control Statement Interpreter (CSI) Component of the Executive. All commands are parsed and syntactically checked by a common parser. Once the command is parsed and determined to be syntactically correct, control is passed to the appropriate Executive component for processing. No security checks are made in the CSI component. All security-relevant checks are made by the Executive routines responsible for the processing requested by the commands.

Executive Control Language (ECL)

The command language of OS 1100 is ECL and consists of commands that begin with an @ or an @@. The command interpreter is an Executive component that reads and interprets these commands and then passes control to the Executive code module which will process the command. There are no direct security concerns at the ECL distribution level, as all parameter validations and access checks are made by the individual command's processing code. There is no security-relevant difference whether a command begins with @ or @@.

The statements that begin with @@ are called transparent control statements. A normal control statement will not be processed until all processing associated with the previous statement is completed. Transparent control statements are entered from demand runs only and are processed immediately. There are two kinds of transparent control statements, ECL and symbiont. The ECL transparent control statements are a specific subset of the full ECL. The same processing occurs as in regular ECL statments. The symbiont transparent control statements allow the user to control the operation of the terminal (e.g., set the timeout value, screen size, scrolling method, or auxiliary device modes).

OPERATOR-CONSOLE KEYINS

Executive (OS 1100) operator keyins are entered from the system console(s) connected to the central system. The operator is given no privileges per se. However, the operator has full control over OS 1100 and, in effect, has all system privileges. A Network Management Services (NMS) console connected to a DCP is used to control the communications configuration of the system.

When the OS 1100 operator initiates batch runs (ST keyin), the user to be associated with the run and the security level are specified by the operator and are verified by the Executive to be a valid combination.[1] No user authentication is performed or required. All operator commands and responses on the main system console are audited.

OS 1100 operator commands are captured and processed by the Console Handler component of the 1100 Executive. The command name is parsed and used in a table lookup to determine the Executive routine to be given control to process the request.

Since the OS 1100 operator can control and modify the TCB, the operator essentially has full access to all protected objects in the system. Electronic identification and authentication of the operator are not required.[2] Interaction between the operator and the 1100 Executive is via operator commands (unsolicited keyins) and responses to system messages (solicited keyins). Unsolicited and solicited keyins are equally security relevant. All messages are received by the Console Handler complex of the 1100 Executive. Unsolicited keyins are denoted by an alphabetic command identifier. This indicates the command and, through internal tables, indicates the Executive component that is to handle the operator request. Solicited keyins are recognized by a message number, which is the first field of the response. This message number identifies the console message to which the keyin is responding, and is used by the Console Handler to route the response to the appropriate requester.

OS 1100 operator commands can be entered from a user terminal. This mechanism (@@CONS) is controlled by privilege. The SSA specifies for each user in the system a console mode with a range of capabilities. This mode is retained in the user-id record in the security database. The console mode determines the specific operator requests that the user can perform. The use of operator requests from a user terminal is audited and the audit record includes the user-id of the requesting user and the terminal from which the request was entered.

---

[1]The audit records of the started batch run will indicate the user-id specified on the run statement.

[2]Physical logs of operator access to consoles must be maintained, however, as per NCSC TCSEC interpretation.

The SSP operator console is not used during normal system operation (i.e., in system mode).[1] Its actions are not audited and they do not affect the running TCB.[2]

The DCP/OS operator console actions are not audited nor is operator logon supported. Since the DCP/OS operator may access TELCON memory, including message buffers, the DCP/OS console will not be used when the system is in B1 operating mode [42].

The TELCON, or Network Management System (NMS), operator can reconfigure the communications lines attached to the system via the DCP. The TELCON operator does log onto the DCP and the security-relevant operator commands are logged [97].

OTHER TCB INTERFACES

Executive interrupts were discussed earlier (page 33, "Interrupts"). The security-relevance of interrupts is that illegal and privileged instructions are handled securely and no memory access is allowed outside a user's storage limits.

MCB (see page 46, "Message Control Bank (MCB)") $Action commands may be issued by the system operator and by TIP terminal users. Only the operator may issue the security-relevant $Action commands.

LBJ instructions are used to reference common-bank subsystems. Upon entry of a process to a subsystem, MAC and DAC checks are made.

User logon parameters are used for I&A and for selection of the mode for TIP.

TIP file execute access controls the execution of TIP programs. File access is controlled by MAC and DAC.

Executive Differences across Hardware Models

The user-visible interfaces to the various hardware models in the 1100 and 2200 family are essentially the same. The specific differences in the user interfaces are identified later in this section, but those differences are not security relevant.

The major differences in the processor types in this family are internal design differences reflecting different speeds, memory and caching architecture, implementation technology, and interconnection mechanisms. The three models share the same system architecture.

---

[1]The console device may be put into system console mode, however. In this role it is not an SSP console.

[2]OS 1100 maintains a record of the initial and subsequent hardware configurations as controlled by the SSP.

To take advantage of the architectural differences and still maintain a single set of source code for the system, the 1100 Executive uses a combination of conditional assembly and configuration parameter data cells to control the execution of code specific to a particular system type and configuration. A configuration parameter data cell is an Executive-maintained data area that contains specific values to denote system configuration information. These cells are initialized to system generation values at system boot time. These values are directly accessible to Executive code to test against specific values and jump around unneeded code. These values are not directly accessible outside of the Executive. A controlled system interface (ER CONFIG$) can be used by the SSA to change the configuration, or by non-Executive users to examine the configuration. While there are exceptions, the configurability of hardware is generally via conditional code generation and the configuration of software is via configuration parameter data cell. This allows software features to be installed and de-installed dynamically while it is generally required to boot a new system for major hardware changes.

Security testing was carried out across all models of evaluated hardware.

Non-Executive Trusted Software

The OS 1100 TCB contains 12 software modules, besides the executive, which are used to perform some security-relevant or protection-critical function. These modules operate with Processor Privilege 2, and do not have any inherent access rights to objects different from the rights of ordinary users. However, these modules are given privileges to bypass the security policy as needed to perform their functions. Privilege assignment to non-Executive software modules implies that these modules must work correctly and only after analysis and testing, by the evaluation team, are these modules accepted as part of the evaluated TCB.

The service or services provided and a capsule description for inclusion in the TCB are given.

Communications Management System (CMS 1100)

CMS 1100 is the Communications Management System for the Unisys Series 1100 systems. It is part of the communications network software which has three major components: TELCON, MCB, and CMS 1100.[1] The communications network consists of all the hardware, firmware, and software which supports communication between remote end users and the Executive. Remote end users are those users that connect through a Distributed Communications Processor (DCP), Workstation Control Unit (WSCU), Communications Line Control Unit (CLCU), or an Integrated Communications Processor (ICP). These four communication units are viewed as front-end processors (FEP) by CMS 1100. While they differ in functionality, CMS 1100 uses common interfaces for all FEPs.

---

[1]These are described in following sections.

41   27 September 1989

CMS 1100 is the software connection between applications on a Unisys host and the TELCON software which runs in the front-end processors. These two products combine to provide the transport, session, and presentation communication layers between host applications and remote end users as defined by the vendor's Distributed Communications Architecture (DCA).

CMS 1100 manages the direct channel physical link between the host and the FEPs. Traffic across the channel consists of control and status information to TELCON and the data associated with the transport, session, and presentation layers.

Remote end users communicate with host applications through sessions which are defined as either demand, batch or TIP. Three components of CMS 1100 control the three classes of sessions: RSDCSU, RSBCSU, and TIPCSU, respectively. CMS 1100 components are called Communication Systems Users (CSU).

The different types of CSU perform quite different functions. However each has the same basic functional structure. For input from a remote end user to a host application, a CSU determines the appropriate session by processing the session protocol. Then the CSU determines the appropriate data representation by processing the presentation protocol, and finally the CSU presents a data unit to the application. Information from an application to a remote end user is simply controlled in reverse order.

CMS 1100 controls printing by remote batch printers. It uses the SMOQUE$ ER to obtain the print file and label from the Executive.

Communication between CMS 1100 and TELCON is defined by DCA protocol. CMS 1100 is part of the TCB because it is essential for the proper routing of communications from remote end users and host-resident applications. Users sign onto the communications system via CMS 1100.[1] CMS 1100 plays a key role in the signon process by: 1) ensuring that new session signons are routed to the appropriate Executive interface, based on application, and 2) causing non-display of the userid/password entered by the terminal user.

Telecommunication Control (TELCON)

TELCON is a software communications networking product which processes on the DCP, WSCU, CLCU, and ICP. CMS 1100 interfaces with TELCON on these four FEPs. TELCON manages all data communications in these FEPs. Any communication between a remote terminal end user and host resident application requires the support of at least one active TELCON and an ac e CMS 1100.

When a terminal attempts to sign on, TELCON selects the network connection based on the user's $$OPEN command. TELCON uses the network communication configuration to map the requested application to a CSU in CMS 1100 to provide transport services.

---

[1]Logon to OS 1100 follows signon.

TELCON is part of the TCB because it determines the type of CSU for an application call and it ensures correct message delivery to the terminal.

Computerized Onsite Maintenance for User Systems (COMUS)

COMUS is the software tool used to generate and configure the operating system software, and to install software products into the system library. It also maintains the integrity of the system library. It is a command-driven application which can be executed in either demand or batch mode. Products are packaged by Unisys so that the distribution medium contains all the information needed by COMUS to install the product in accordance with the site's security requirements. COMUS uses this information to control common bank access. This means that common bank products are installed as protected subsystems, including the use of security gate banks. The proper installation of common banks ensures access control within the system. COMUS is included in the TCB because of its role in configuring the rest of the TCB software.

File Administration System (FAS)

FAS is the system software component used to perform administrative functions on the file system [4]. FAS provides backup, archival, tape management, and reporting functions. FAS operator functions can be used by either an SSA or any user with or without privilege. In order to restrict a user to only the functions that should be performed, FAS provides two modes of operation; "Full Privilege Mode" and "Not Full Privilege Mode". In Full Privilege Mode, FAS performs privileged Executive Requests.[1] To ensure this mode is properly invoked, FAS compares the privileges allowed the invoking user with the privilege required to perform the FAS request. If the invoking user does not have the appropriate privilege in the user security profile then FAS sets the mode to "Not Full Privilege Mode" and restricts the set of privileges to those allowed a user without privilege. The minimum security privileges needed by the user for FAS to process a request in Full Privilege Mode are: SSBPFC, SSADID, SSDRG, SSLASH, SSBRWK, SSBROD, SSBKUP, SSDNK, SSPLACE, SSCSPF, SSCRCL, SSBAFC, SSBYCL, SSCCL, SSBYCOMP, SSCHCOMP, SSBYPASSOWNR,SSCHCACR, SSCHDR, SSMROOC, SSREMFILOWNR, SSBYTIMUPDAT, SSFDELE, and SSDBACK.

BACKUP AND RECOVERY

FAS provides a tool for any user to create backup copies of mass storage files to tape and to retrieve back up files from tape and return them to mass storage. File attributes remain the same throughout the backup process. FAS allows a user with full FAS privilege to back up any selected file identified in the MFD. The SSA may choose to backup the entire MFD or just selected files.

---

[1]FAS does not have its own privileges. It operates under the privileges of the user. It must be in the TCB to ensure that it does not abuse privileges.

To ensure that a file is not downgraded, FAS compares the security attributes of a file being restored with the security attributes of the current, existing mass storage version of the file (if it exists). If the security attributes of the two versions do not match, an error message is printed and the file is not restored.

ARCHIVAL OF MASS STORAGE FILES

FAS allows the SSA to specify files that should be archived to tape. When ·. file is archived, it is copied to tape, its MFD entry is removed, and an entry is placed in the Archive Directory. An archived file cannot be referenced directly on the system until it has been retrieved by a user with full FAS privilege.

The system provides a virtual mass storage capability so that whenever available physical mass storage space is less than a preset level, the Executive calls FAS to unload mass storage files to tape. FAS uses a predefined algorithm to determine appropriate files to unload. The record of the file being unloaded is retained in the master file directory (MFD). Subsequently, if a user attempts to access such an unloaded file, the Executive will set an interrupt and call FAS which will retrieve the file, and identify within the MFD entry that the file now resides on mass storage. This virtual mass storage capability is transparent to the user and all file attributes remain unchanged.

AUTOMATED MANAGEMENT OF FAS TAPES

FAS provides a Tape Pool Manager for managing the tapes that it handles. The Tape Pool Manager allows a user with full FAS privilege to:

- add and delete tapes to and from the pool.

- mark and reserve tapes within the pool.

- avoid accidental re-usage of a tape that contains current information.

- allow prescreening of tapes that will be used.

These functions are provided in order to relieve the SSA of the procedural difficulties in management of backup tapes.

ACCESSING AND REPORTING ON THE MFD AND ARCHIVE DIRECTORY

FAS provides a variety of searching and reporting capabilities for the file system. It allows the SSA to select files based on attributes of the file such as its project-id, name, whether it is loaded or not, and its backup date. FAS also allows selection of files based upon any combination of attributes to help the SSA create a viable backup procedure.

Integrated Recovery Utility (IRU)

This component, part of the Integrated Recovery Version III System, provides database and transaction message recovery, either offline or in the background. It makes tape backup copies of database files and reconstructs these files from the tape backup copies and information from the audit trails, when needed. It processes the audit files and therefore must run at a system-high level.

The IRU is executed in a runstream using the IRU ECL command. To invoke some of the functions of IRU, the user must possess some combination of the following security-relevant privileges: SSBYCOMP, SSCCL, SSBYPASSOWNR, SSSSCALLANY, and SSADID. The security-relevant ERs needed are MSCON$, DREG$, EACQ$, BDSPT$, and MODPS$ (see page 64, "Privileges"). Because it runs with bypass privileges, the IRU can access and utilize TIP and Universal Data System (UDS) database and message files, as well as the Exec Audit Control Interface File (which contains audit trail linkage information). Also, it invokes UDS and the Message Control Bank (MCB) (see below) to aid in the recovery of database files and messages, inasmuch as IRU does not contain all the information on the structure of the units it recovers. It has interfaces to the Executive components Exec Step Control and Audit Control. Step Control directs recovery actions in a DBMS environment. Audit Control maintains audit trails for recovery and other purposes (e.g., performance analysis).

IRU is included in the TCB because of its pervasive access to critical system files.

Universal Data System (UDS)

UDS is a group of software products brought together to help manage the flow of database information in the system. The products include Universal Data System Control (UDSC), Data Dictionary System (DDS 1100), Data Management System (DMS 1100), Relational Data Management System (RDMS 1100), Relational Syntax Analyzer (RSA), Define File Processor (DFP), and Shared File System (SFS 1100). UDSC, the UDS database control software component, coordinates operations among individual software products and application programs. UDSC handles data at different levels by defining an application group at each security level with its own data banks. A copy of UDS exists for each application group, i.e., database environment, on a system. Each database environment will be at a single security level. Each UDS copy is protected by a separate subsystem.

IRU accesses UDSC when performing database recovery. Since UDS must have access to audit trail data from an IRU bank, and that data may be multilevel, UDSC must be trusted to behave properly. It is therefore included in the TCB to ensure the proper isolation of different levels of data. The rest of UDS, including the DBMS packages, is not included in the TCB.

No special privileges are needed by UDSC. It uses the AUDIT$, DMABT$, and MQF$ ERs (see page 64, "Privileges").

Message Control Bank (MCB)

The MCB is the message control component of the Series 1100 Integrated Recovery (IR) and TIP environments. Each IR or TIP application includes a unique MCB, comprising the MCB subsystem and message retention files, to provide message staging, message auditing, message queuing, and message recovery. For TIP sessions, CMS 1100 interfaces with the MCB subsystem to acquire output messages and pass input messages. TIP programs call the MCB to obtain input messages and pass output messages. Batch-connected programs also call the MCB to acquire input and pass output messages. The MCB interfaces with the Executive to track the state of messages and transaction programs.

One MCB subsystem is created for each active IR or TIP application, to a maximum of nine simultaneous applications. To be active means that an IR application has initialized an MCB subsystem of common banks and files. The interface between MCB and CMS 1100 is always initiated by CMS 1100, which enters the MCB common bank. The TIPCSU of CMS 1100 uses the host-resident application number to determine the BDI (see page 8, "C-Series Addressing") of the MCB common bank which will process messages for the session after it is successfully opened. Once initialized, the host application remains active, memory resident, and real-time, but in a waiting state. Mass storage files created by MCB are TIP/DMS (Data Management System) files which are used for message retention. TIP/DMS files are controlled by the standard 1100 Executive file-protection scheme.

Peripheral Control Software (PERCON)

PERCON controls all printers connected to the system via I/O channels. Originally, support for these devices was provided in the OS 1100 Executive but support for local printers is now being provided by PERCON. The user-id for PERCON must be a system-high user (full clearance range and all compartments), and be assigned the privileges SSCCL, SSSMOQUE, SSCONSOLE, SSLOGER, SSBPFC, SSBRWK, SSADID, SSBAFC, SSBYCOMP, and SSBYPASSOWNR. The MODPS$ and SMOQUE ERs are also used. PERCON requires these privileges to be able to have unrestricted access to a user's data in order to print the user's files, and to communicate with the Executive and the operator.

PERCON uses the SMOQUE$ ER to assign files to be printed and to control print queues. Its other privileges give it unlimited access to print files. Print queues are protected by the Symbiont Interfaces (SYI) of the Executive. Separate activities in PERCON read print files and send output to the printers. When PERCON executes the SMOQUE$ ER, the Executive SYI provides security label information to PERCON and a file containing data to be printed is assigned to the PERCON background run. PERCON maintains the label in a common bank during printing of the file. PERCON carries out banner and trailer printing and every-page labeling.

There may be a maximum of two PERCON runs active in a system, and each of the runs can control up to ten devices. The runs share one common bank, and each device has its own data bank. The print queue is maintained in the system file GENF$. PERCON ensures that residue from a print file is not printed on subsequent uses of its print file space.

Site Management Complex (SIMAN)

SIMAN is the high-level interface into the security database (see page 29, "Security Database"), which provides a full-screen interactive interface that uses 1100 Executive ERs at a primitive level to carry out the requests of its users. SIMAN is installed as a trusted subsystem and has the full range of trusted subsystem security attributes. All references to SIMAN pass through the subsystem gate so that all levels of users can access SIMAN properly. Also, hardware protection mechanisms ensure that control is transferred to SIMAN's correct entry point. If SIMAN is called illegally, it will abort in a manner that protects all sensitive data.

When SIMAN is called, it reads the user's user-id records from the SYS$*TSS$FILE and SYS$*SACRD$ files and checks the run's Program Control Table (PCT) to determine the attributes of the user. SIMAN determines whether the user is an administrator or subadministrator, if the run is under the Master Account (see below), and the minimum and maximum security attributes of the user. This information is used by SIMAN to identify the functions that the user is allowed to do and the full screen menu is built to give only those options [4].

SIMAN Administrators

Certain user-ids on the system are designated as either a SIMAN administrator or subadministrator. This designation allows them capabilities that other users do not have in SIMAN.

A SIMAN administrator has access to all information in SYS$*TSS$FILE. An administrator can perform all user-id installation, user-id maintenance, and define subadministrators. A SIMAN administrator is identified as such by a field in the user-id security record.

A SIMAN subadministrator has access to any information in SYS$*TSS$FILE that the subadministrator has created. A user-id designated as a subadministrator can install user-ids and perform maintenance on those user-ids created by it. A subadministrator may or may not be able to define other subadministrators, depending on whether that capability was given to the subadministrator's user-id.

SIMAN administrators and subadministrators need the following privileges as well as their security-record designations in order to fulfill their defined functions: SSCSU (create security records), SSCRCL (modify record clearance level), SSCHCOMP (change compartment set within the user's maximum set), and SSCSPF (modify special flags). To set the 'ALL' flag, a(n) (sub)administrator must have access to all compartments. SIMAN

administrators and subadministrators also must have ER ACCNT$ to bypass account number controls.

Master Account

The Master Account is an account designated as such in the SYS$*ACCOUNT$R1 file. Only when a run is under the Master Account may updates be made to the SYS$*ACCOUNT$R1 file. Use of this account number is password protected. Otherwise, only information pertaining to the current account may be read. Because the ability to run in real-time mode is controlled through SYS$*ACCOUNT$R1, this has extra security considerations (see page 62, "Access to Real-Time Words").

System Calls (ERs) Used By SIMAN

SIMAN uses 1100 Executive ERs to satisfy the requests made by users. Generally, SIMAN just provides a user-friendly interface and composes the arguments it gets from users into the proper form for the ER it needs to call. The exception to this rule is ER USER$, which provides only one check as described below.

Updates Using ER USER$

All updates to the SYS$*TSS$FILE (logon information) are handled by SIMAN using the ER USER$. SIMAN is responsible for checking all updates to ensure that the user is not trying to perform file changes beyond what the user-id allows. SIMAN allows designated administrators and subadministrators the capabilities explained previously (see page 47, "Site Management Complex (SIMAN)"), but unprivileged users may only read information from their own user-id record. SIMAN can return all information requested on any user-id record the executing user has access to, except for the password. Passwords are never returned from SIMAN.

To ensure that no outside users call the ER USER$ and make changes to the SYSTSSFILE without SIMAN, the 1100 Executive will only accept the request if USER$ is called from a specific BDI (see page 8, "C-Series Addressing"). When SIMAN is installed on the system and is occupying the reserved BDI , no other program may call USER$. Additionally, the USER$ ER is a controlled ER whose privilege is only given to the SIMAN common bank subsystem.

Usage of Normal ERs (not USER$)

SIMAN calls several ERs besides USER$ to perform functions. Below is a list of them and the services they provide.

- SPRNT$ - used to display user-id and ACR records from the SACRD$ file.

- SREG$ - used to create user and ACR records in SACRD$.

- SUMOD$ - used to update the SACRD$ portion of the user-id record and the cataloged file security records in the MFDF$$ file.

- SDEL$ - used to delete user-id and ACR records in SACRD$.

- SUVAL$ - used to determine user access to other user-id records and cataloged file security records. Checks user against record's clearance level, compartment set, and Access Control Record.

- STAB$ - used to maintain the list of secured ERs and privileges. These lists are used to create the user-id record maintenance screens.

- MSCON$ - used to retrieve security information for the cataloged files from the lead item of the Master File Directory.

- SCDTL$ - used to produce listings of the compartments accessible to the executing user. SIMAN uses the list of the maximum set to edit user-id record updates.

- SCDTA$ - used by privileged users to update the Compartment Definition Table. SIMAN supports Modify, Update, Rename, and Decontrol for users with appropriate privilege.

- SCLDT$ - used by SIMAN to maintain the Symbolic Clearance Levels. SIMAN access to this ER is also controlled by a privilege.

- ACCNT$ - performs all updates to the ACCOUNT$R1 file. This includes accounts, user-id attached to accounts, and Quota Sets. The user must be running under the Master Account in order to perform updates. Users not running under the Master Account can only read information pertaining to their current account. The ACCNT$ ER is also a controlled ER whose privilege is required to access the SYS$*ACCOUNT$R1 file.

- CONFIG$ - used by SIMAN to return information about the system configuration. This includes information about the level of security or accounting, the type of tape labels to create, and the version of the TSS$FILE. SIMAN uses CONFIG$ for information only, and never updates actual system configuration.

For more detailed information on each of the ERs, see the ER Programming Reference Manual [81].

System Support Processor (SSP)

The SSP is a system with its own processor which is used to control system partitioning, initialization, and fault recovery. On the 1100/90, the SSP can be used to partition the hardware complex into up to four independent systems. The SSP provides the hardware

gates to ensure isolation of such systems down to the I/O subsystem level. Such partitioned systems are not a part of the candidate class B1 evaluation. Once the system is initialized (booted), it operates in system mode. It must be protected from SSP activity and from initial program loads from the SSP panel.

During boot by the SSP, the OS 1100 Master Configuration Table records the available hardware components. After the system is up (system mode), the audit trail records configuration changes as either hardware fault or operator keyin entries [95]. The SSP operating system, the SSP Supervisor, communicates with OS 1100 via Exec Link Functions (ELF). The SSP Supervisor makes use of no OS 1100 privileges or ERs [4].

Tape Labeling System (TLABEL)

The Tape Labeling System (TLABEL) is a utility used for the maintenance of labels on tape volumes. It is composed of the TLABEL processor and a set of predefined runstreams [4]. TLABEL can be executed interactively or may be started as a batch job using one of the predefined runstreams. TLABEL provides the capabilities to create, remove, and print tape labels.

TLABEL performs no security validations for user requests. It composes packets and makes system calls to the 1100 Executive. The 1100 Executive determines if a user has the privileges required to perform a specific function within its ERs.[1]

System Calls (ERs) Used By TLABEL

The following are the ERs called by TLABEL to satisfy user requests:

- SPRNT$ - used by TLABEL to verify the existence of an Access Control Record (ACR) before putting the ACR name into a tape label.

- SUMOD$ - used to modify the security attributes in the MFD for cataloged tape files.

- SUVAL$ - used to verify access to ACRs and cataloged tape files.

- SCLDT$ - used to convert numeric clearance level to its symbolic equivalent.

- MSCON$ - used to read security information for cataloged tape files.

- TLBL$ - used to write the volume header label (VOL1) on a tape volume.

---

[1]TLABEL does not have its own privileges. It operates under the privileges of the user. It must be in the TCB to ensure that it does not abuse privileges.

- TVSLBL$ - used to write a tape volumes's security attributes in the VOL3 record.

- SCOMCNV$ - used to convert the compartment set bit mask to the symbolic names.

Disk Preparation Routine (DPREP1100)

DPREP1100 is the system software for preparing disk packs and for identifying and managing defective areas on disks. DPREP1100 is started by the console keyin ST DPREP.

The user-id of the DPREP run must have the SSADID privilege which allows absolute device assignment. The SSADID privilege grants DPREP1100 the ability to read any user data with ER IOAID$, so the run must operate at system high (level 63, all compartments) in order to ensure no declassification of data occurs.

Whenever DPREP1100 is executed, it reads input from the runstream or input device, and directs output to the user's print file or to the system console, depending on its volume and importance. However, before the 1100 Executive allows absolute assignment[1] of a previously prepared pack on the system, the console operator must respond to a console message. Thus, possible access to user data on a prepared disk pack is only possible through the explicit action of a console operator and even then only to a privileged user.

Excluded Items

There is major software and hardware that Unisys provides and markets but which Unisys has agreed cannot be included in the Class B1 configuration. Following is a list and a principal reason the products or features are excluded:[2]

DDP not	Uses write-enabled common banks, file labeling is maintained.
OFISLINK	Does not provide mandatory access controls or labels.
NPE (New Programming Environment)	Needs modification for B1
Use of @FILE or @ENDF	Both allow a user to access any file.

---

[1] i.e., by device identifier

[2] This list is provided to help delineate the TCB but is not a comprehensive list. Any system with privileged code that is not identified in this report violates the evaluated rating.

Use of @@TM	Would allow any user to send messages to another.

Multiple Host File

Sharing (MHFS)	Needs modification for B1
High-Volume TIP	Reuses writable banks

Multiple-Initial TIP transactions
Reuses writable banks

Attached Processors
Needs modification for B1

SX1100 (Unix)	Needs modification for B1
IMS-1100	Needs modification for B1

2200/200 Workstation as a Console
Needs modification for B1

Remote System Support (RSS)
Needs modification for B1

Checkpoint/Restart (CKRS)
Needs modification for B1

Monitor Services Control Program (MSCP)
Needs modification for B1

OSAM Re-entrant Mode
Needs modification for B1

Transaction Processing Auditing System (TPAS)
Needs modification for B1

Auxiliary Terminal Printers
Output not labeled

TCB Protection Mechanisms

OS 1100 has mechanisms by which it provides protection for its TCB components. An audit log of security-relevant events is kept by OS 1100 for use by an authorized administrator. Interfacing with the Executive is limited to defined methods, so as to protect its integrity. The system protects against inadvertant disclosure through object

reuse. Mandatory access control is implemented using a hierarchical and compartmented labeling policy consisting of sixty-four levels and up to thirty compartments. Also, discretionary mechanisms are available to finely control access to objects within a security level and compartment set.

Access Control

The 1100 Executive implements access controls between all of its subjects and objects. These access controls uphold the OS 1100 security policy by giving access to authorized users and denying access to unauthorized users. Information needed to make access decisions is kept in the security database (see page 29, "Security Database"), the Master File Directory (MFD).[1] For a description of the ERs used to access security information, see page 47, "Site Management Complex (SIMAN)".

The files comprising the security database are catalogued (registered in the MFD) during a boot of the system with the 1100 Executive designated as their owner. The files are exclusively assigned (opened) by the 1100 Executive and, as an extra precaution, the names of the files all contain six characters that are impossible for a user to express to the system. Therefore, no user can assign (open) the security database files and no user can perform I/O to the security database. The only access to the database is through ERs.

Accessing the Master File Directory (MFD)

The MFD is stored in a temporary[2] system file called SYS$*MFDF$$. Through ERs, it is accessible to anyone on the system. If a new file is created or some of the characteristics of a file change, then the MFD is written to. The data in the MFD can be read by anyone.

Accessing the Quota Data File

The Quota[3] system on OS 1100 keeps its data in a file called SYS$*ACCOUNT$R1. The ability to run with real-time priority is controlled through the settings in this file. Only when a run is under the Master Account may updates be made to the

---

[1]Security data exists in Executive internal structures also, but the information always originates from those files mentioned here (i.e., security database, etc.).

[2]A temporary file is a file that does not have an MFD entry and only exists for the life of the run that uses it. The run in this case is from system boot to system stop. Information to build the MFD is stored on mass storage along with the files, enabling the TCB to restore the MFD after the system has beeb taken down.

[3]The "Quota system" is a term used to refer to controls imposed by using accounts and account information. For example, DAC can be controlled on the basis of whether a user is logged on under a certain account. Users log on under a certain account during the system identification and authentication process.

SYS$*ACCOUNT$R1 file. Otherwise, only information pertaining to the current account of a run may be read.

Accessing the System

When an SSA registers a user-id with the system, it is assigned one of three logon modes along with a type of system access.

SYSTEM ACCESS TYPES

End users access the system by initiating a run or a session. Runs are batch or demand jobs and sessions are transaction processing periods. A run is defined [16] as a "sequence of tasks that are linked together to form a self-contained unit of work." A task is defined as "a discrete processing step in a run, involving the execution of an absolute element (either a system processor, i.e., a system program, or a user program). Generally, each ECL statement in a runstream represents a task." A user-id may be configured to have any combination of the three types of system access: TIP, demand, and batch.

If the user-id has TIP access, the user can log onto a TIP terminal session. If the user-id has demand access, the user can log onto a demand terminal and begin a demand run. Batch access gives the user-id the ability to start a batch run using any of the methods mentioned on page 70, "Batch Runs." Users with either damand or batch access may connect to a TIP session.

LOGON MODES

The three types of logon modes are basic, run, and execution mode. These are equivalent so far as security relevance is concerned.

If a user-id is configured for basic logon mode, the system requires a user to enter an @RUN statement immediately after logon. In run mode a user does not have to enter an @RUN statement to start a demand run, the system does that for the user. To bypass the automatic @RUN statement, the user enters an asterisk (*) before the user-id when responding to the logon request. Execution mode is just like run mode except that the user does not have the choice of entering his own @RUN statement. If an asterisk is entered before the user-id, it is ignored by the 1100 Executive. Execution mode implies that remote batch runs cannot be started for a user-id, since these require an @RUN statement.

TERMINAL LOGON PROCEDURES

When using a terminal to access OS 1100, the logon procedure will be initiated as either a TIP logon or a demand logon. The procedure is very similar for both types. The third type of access to the system, batch, is discussed in the following section, "Batch Runs."

To log onto OS 1100, a user must identify himself to the system with a unique user-id and authenticate that identity with a password that is valid for the user-id. The user must

also have a clearance level and compartment set at which he wishes to execute. The clearance level and compartment set entered by the user are validated against the capabilities in the user-id security record. Another security attribute that a user must have is an account number [91]. If a solicit flag is set for the user, he will also be prompted for a project-id. If a project-id is not given, the default is QQQ$. The account number and project-id supplied by the user are validated against the pre-defined allowable values in his user-id security record. QQQ$ is always a valid project-id.

BATCH RUNS

Batch runs may be started a number of ways. A batch run may be started using the @START statement from a demand or another batch run, it may be started using the ST console keyin, or it may be submitted to a card reader. The password is not given in a batch runstream which resides in a file. Only in the case of card reader input is the run's password included in the runstream. Two statements, @FILE and @ENDF, are not allowed in a runstream because they do not conform to the security policy. If either of the two statements is detected in a runstream, the run is not initiated.

Regardless of the method for initiating a batch run, all the information that would be specified during a demand or TIP logon must be supplied. This information includes user-id, clearance level, compartment set, account number, project-id, and password. When a batch run is started from a demand terminal, the I&A information for the batch run is taken from the starting demand run.[1] The first statement of a run is always the @RUN statement, which can specify user-id, account number, and project-id. The @PASSWD statement specifies password and security level if the runstream is entered from a card reader.

The U option for @START and the ST keyin allows a run to be initiated with someone else's user-id. This ability is limited by the fact that to use the U option, the starter (user who starts the run) must have execute access to both the file containing the run stream and to the started run's user-id record.[2] The started run's user-id must own the file containing the run stream. The security attributes of the starter are used for the run, but they must be in the range of the owner of the file containing the run stream. The privilege set of the started run is that of the user-id specified in the stored runstream. The audit records contain the user-id that started the batch run. All following audit entries for the batch run have the user-id of the owner of the file containing the run stream.

A starter who is cleared to system high may start runs at system high while executing at less than system high if he possesses the privilege SSSTRTZOPT. This mechanism is activated by the Z option on the @START control statement or by the ST console keyin.

---

[1] A password is not needed for the ST console keyin.

[2] Provided by the DAC mechanism.

Accessing Subjects and Objects

One subject type, the subsystem, is defined by the OS 1100 security policy as are three object types. The object types are files, tape volumes and subsystems. For more information on policy defined subjects and objects, see page 20, "Definition of Subjects", and page 25, "Definition of Objects".

Only for the purpose of discussing access controls, TIP-defined terminals will be discussed as storage objects.

Mandatory Access Control

The 1100 Executive implements mandatory controls based on a clearance level and a compartment set. Together, the clearance level and compartment set form a security level, defined for both subjects and objects.

OS 1100 has one external interface to the Executive security validation routines, the SUVAL$ ER. This ER provides a common entry point to security validation for all callers outside the central security complex. SUVAL$ determines whether or not a subject has access to an object, whether or not a subject may execute a given ER, and whether or not a subject has a given privilege. ER SUVAL$ provides a mechanism to perform security validation, but it does not itself enforce the results of that validation. To use SUVAL$, the caller must pass a valid packet of input parameters, as described in the TFM [42].

Clearance levels categorize users, data, and resources in the system in a hierarchical fashion. Clearance levels can range from 0 to 63, 63 being the highest. Level 0 may be read by all subjects. Level 1 and higher may only be read if the subject's range includes the level. An installation can configure a system to use less than the full 64 clearance levels and each level used may have a symbolic label. Note, however, that level 63 must be defined or no access will be allowed to anything requiring system-high access. Symbolic clearance levels are names which correspond to numeric clearance levels for a particular installation.

Compartments categorize users, data, and resources in the system in a non-hierarchical fashion. An installation can choose up to 30 compartment names at one time. When compartments are defined for a particular installation, the 1100 Executive decides which bit is to correspond to a symbolic name. In addition to this, a version number is associated with each bit so that the TCB can tell if a compartment bit in the security level of an object has been redefined or decontrolled in the system. If the compartment bit in an object's label is no longer a valid definition, the bit is not used in making access decisions.

Discretionary Access Control

The 1100 Executive implements discretionary controls based upon an access list field in the security record. The field may contain blanks, a null value (0), or the name of an Access Control Record (ACR). If the field contains a null value, the object to which it

applies is private to the owner. The owner of a file is the user who created it. If the field contains blanks, the object to which it applies is public to all users which pass mandatory access controls. If a name is in the field, discretionary access is controlled on the basis of the ACR indicated by the name. If the ACR does not exist, access defaults to private.

ACCESS CONTROL RECORDS (ACR)

ACRs implement a discretionary protection mechanism which allows four access types: read, write, delete, and execute [17]. Execute is interpreted as enter access for subsystems. ACRs can specify which user-ids, project-ids, and account numbers can access an object or user-id security record. ACRs can also restrict the time of day the object may be accessed. Entries in ACRs include one or more Access Control Functions (ACFs). Each ACF begins with the access types to which this ACF applies and how many arguments are in the ACF. Each argument is a three-word entry consisting of (1) an object name or value to be compared using the logical relation specified in the argument and (2) a logical connective showing the logical association with the next argument(s) in that ACF. ACRs are developed using SIMAN screens or the ER SREG$. The logical syntax as specified by the user in a free-form manner is checked by SIMAN and errors are identified.

The following example allows JJones, James, and Baker while working under the CPES account to have read and execute access. Baker's access is available only between 8:00 am and 2:00 pm: "READ/EXECUTE USERID EQ JJONES AND ACCOUNT EQ CPES OR USERID EQ JAMES AND ACCOUNT EQ CPES OR USERID EQ BAKER AND ACCOUNT EQ CPES AND TIME WR 0800-1400". Each user has a quota of ACRs which is set by the SSA.[1] After an ACR is created it cannot be modified, so extreme care must be taken when creating an ACR so that it does its job properly the first time. A particular ACR can be used on more than one object. Note that the ACR must exist before its name is put into an object's access list field. Deleting an ACR does not decrement the "ACRs owned" count. So, theoretically, a user may not own any existing ACRs, have a quota (e.g., five allowed), and still not be able to create an ACR to protect his files because he has created and deleted five ACRs in the past. (In practice, the SSA could and should reset the user's ACR count). The owner of the object or user-id security record must match the owner of the ACR. ACRs are private to their owners.

Read-only and Write-only

The read-only/write-only mechanism augments the protection provided by the access list field. This mechanism is useful for extra protection of files known to be needed as write-only or read-only; however, the concept of read-only/write-only is not used to enforce the OS 1100 security policy. This means that read-only/write-only mechanisms need never be used for sufficient discretionary protection of objects. Since the functionality does exist, the total possible discretionary controls on the system is the

---

[1]The default mumber of ACRs for a user-id is zero, however zero should not be the amount allocated. The TFM provides general guidance on setting the number to an appropriate value for a user.

combination of the access list mechanism and the read-only/write-only controls. Read-only and write-only protection only protects a single version (F-cycle) of a file, while DAC protects all F-cycles of a file. Read-only and write-only files may be deleted by a user with read or write access, respectively, plus any other required access.

Access to Subsystems

A subsystem as an object has both discretionary and mandatory security information determined by its owner's user-id security record. Discretionary access is determined by the access list field in the user-id security record. If the field contains a null value, the subsystem is private to the owner. If the access list field contains blanks, the subsystem is public to all users passing mandatory access requirements. If the field indicates an ACR name, the ACR has only one type of access that is defined for subsystems: enter access. Enter access is implemented as 'execute' access in the ACR.

When one subsystem (the source subsystem) calls another subsystem (the target subsystem), the 1100 Executive verifies that the source subsystem has enter access to the target subsystem. Additional security checks are performed once enter access has been validated.

The rules concerning the necessary security attributes for referencing a subsystem are contained in the TCB Design Documentation [4]. These transition rules were stated in the section beginning on page 23, "Trusted Subsystems".

Access to Files

Discretionary access for a file is defined by its access list field. The access list field of a file is filled with a null value by default when it is created. This means that the default discretionary protection for a file is private to owner. If the access list field contains blanks, the file is public to all users passing mandatory access requirements. For finer control, an ACR may be attached to the file. ACRs controlling access to files have meaning for all four defined types of access: read, write, execute, and delete.

With respect to mandatory access, whenever a file is created it gets its clearance level and compartment set from its creator's executing clearance level and compartment set. After initial file creation, mandatory access to a file can be defined by two sets of rules, one set governing read and execute accesses and one set governing write and delete accesses.

The mandatory access rules allow read and execute access if:

1. The executing clearance level of the user is greater than or equal to the clearance level of the file and the file's clearance level lies within the clearance level range of the user; or the clearance level of the file is zero.

and

2. The executing compartment set of the user contains the compartment set of the file. Note that NULL is always a member of a compartment set.

The mandatory access ru es allow write and delete access if:

1. The executing clearance level of the user equals the clearance level of the file.

and

2. The executing compartment set of the user equals the compartment set of the file.

Besides its other security attributes, a file has some special flags of security interest. The 'security disabled' flag allows the SSA to make the file inaccessible to requesters who are not executing at security-privileged system high. This is useful when importing unlabeled files. The 'notify security console' flag or the 'notify system console' flag cause a message to be sent to the system console whenever the file is accessed. These features may be used by the SSA to monitor user activity. Lastly, the 'ALL' flag signifies that the file belongs to all compartment sets. This flag provides an efficient means for the 1100 Executive to check for access to the full set of compartments.

Protection of files is also provided through assignment with exclusive use. This restricts access to a file to the run (see page 53, "Access Control") in which the file assignment is made. The 1100 Executive may have files exclusively assigned to its own run, to protect these files from unauthorized access.

Access to Tape Volumes

Physical reels of tape exist as tape volumes and have labels (as described below) written on them. Tape volumes may or may not appear in an MFD entry. If an entry exists in the MFD, it appears as a reel number within an entry for a tape file. The tape volume is accessible via the tape file name in which the volume's reel number appears. Access to tape files is handled the same as for mass storage files and is described on page 58, "Access to Files". When a tape volume has an MFD entry, it may be accessed directly using ECL statements. The TCB need not even be aware that it is a cataloged tape. Access to a tape volume through a tape file name in the MFD entry ensures that the MFD label and the label on the tape volume are the same and makes it unnecessary for the tape's characteristics to be re-specified on the @ASG statement (see page 35, "Executive Interfaces"). Access to a tape volume directly only uses the label on the tape volume.

---

[1]ALL means all defined compartment sets at the time of the check. If the overall definition of compartments changes in the future, ALL covers those too.

Discretionary access for a tape volume is defined by its access list field. The access list field of a tape volume is filled with a null value by default when it is created. This means that the default discretionary protection for a tape volume is 'private to owner'. A tape volume may be made public or an ACR may be attached using TLABEL. ACRs controlling access to tape volumes have meaning for two defined types of discretionary access: read and write.

If a tape volume is prelabeled[1], its security attributes are specified on the label; otherwise it gets its clearance level and compartment set from its creator's executing clearance level and compartment set. After initial creation of a tape volume label, mandatory access to a tape volume can be defined by two sets of rules, one set governing read accesses and one set governing write accesses.

The mandatory access rules allow read access if:

> 1. The executing clearance level of the user is greater than or equal to the clearance level of the tape volume and the tape volume's clearance level lies within the clearance level range of the user; or the clearance level of the tape volume is zero.

> and

> 2. The executing compartment set of the user contains the compartment set of the tape volume. Note that NULL is always a member of a compartment set.

The mandatory access rules allow write access if:

> 1. The executing clearance level of the user equals the clearance level of the tape volume.

> and

> 2. The executing compartment set of the user equals the compartment set of the tape volume.

Because tapes, as opposed to files, are removable storage media, the following handling procedures are established. All tapes must have an external label. The operator attaches this label when a tape is prelabeled or, if it is not prelabeled, when a user first writes to it.

For protection, all tapes should be prelabeled with at least a VOL1 format specifying owner of the tape. A VOI 3 format can be prelabeled specifying security attributes also, or a VOL3 will be written when a user first writes to the tape. At the time of the first

---

[1]at the installation, e.g., by an authorized operator

write, a message will be sent to the operator console providing information for affixing a proper external label.

TIP Session Access

When in a TIP session, accesses are made to objects referred to as TIP files. TIP files exist in 1100 Executive files and support the system security policy. Note that TIP processing may access any of the defined objects on the system, not just TIP files. Also, it is possible for a TIP session to send a message to a particular TIP terminal if it has the privilege to do so (SSTIPSENDMSG).

TIP Files

TIP files are grouped for efficient access. All files in a group have the same security attributes, both DAC and MAC. These TIP file groups are referred to as bundles. An 1100 Executive container file is first catalogued into the MFD and the container file is then registered (ER FC$REG) [57]. This assigns the file exclusively to OS 1100 for use with TIP File Security. Finally, individual TIP files are reserved within the registered file. So TIP files are actually files within a file, and inherit security properties from the container file. If a container file was restricted to read-only or write-only in the MFD, this restriction will be ignored when the file is registered with TIP.[1] Once a file is registered with TIP, it can only be accessed through the TIP environment. A TIP session's access to a bundle is validated and audited at least once per session, at first access.

TIP Terminals

A TIP session can send a message to a particular TIP terminal if the user-id identifying the session has the SSTIPSENDMSG privilege.[2] Since a privilege is required to write to this device, it is not a named object and therefore requires no discretionary access controls.

Mandatory access controls are implemented for the device by associating the terminal with whatever user is logged onto it. The MAC attributes of the sending session are validated against the MAC attributes of the session at the receiving terminal, and the message is not displayed on the receiving terminal if the attributes do not meet the following access controls:

1.   A TIP user can only receive TIP messages after passing clearance level validation.

---

[1]Read-only and write-only restrictions are not necessary under the OS 1100 security policy. They are mentioned in this report because they further restrict the DAC policy.

[2]Only those users whose application requires printing to a TIP terminal would need this privilege.

2. A TIP user can only receive TIP messages after passing a compartment set validation.

3. A TIP user can always see unclassified public TIP messages, i.e. those with clearance level 0 and the null compartment set.

4. Private TIP messages are only viewable by the originator and a user with the bypass ownership privilege SSBYPASSOWNR.

5. Receiving and/or sending sessions running with privilege SSCCL can bypass the TIP message security clearance level test within their clearance level range.

6. Receiving and/or sending sessions running with privilege SSBYCOMP can bypass the TIP message security compartment set test within their compartment set range.

7. Receiving sessions with or without other privileges can only receive messages if their access controls are equal to those of the sending session

Access to Real-Time Words

The 1100 Executive has a certain number of words in a system table (MCT Application Area) that can be shared arbitrarily by any program executing in real time. The ability for a program to execute in real time, and thus access the shared table, necessitates the inclusion of any real-time program in the TCB. Only those real-time applications evaluated as part of the TCB are allowed on the Class B1 system. Real-time mode is an account or user-id attribute that is controlled as part of the Quota mechanism (see page 53, "Access Control"). CMS and PERCON, for example, are parts of the TCB that execute in real-time mode.

Human-Readable Output

Human-readable output is labeled through the TCB components PERCON and CMS. Each submitted print file is preceded by a banner page and followed by a trailer page that each have the file's sensitivity labels displayed. By default, the file's sensitivity labels are also printed at the top and bottom of every printed page. The banner and trailer pages cannot be overridden, but the sensitivity labels on each page can be suppressed by using the 'N' option on (@)@SYM. This suppression of every-page sensitivity labels is audited. Sequence numbers are generated for the printed output, with the first one in a queue being random.

Auxiliary printers on terminals have been deconfigured for B1 because they do not provide output labeling.

Audit

The 1100 Executive creates and maintains an audit file of security-relevant events. All of these events are always audited. Auditing by security level is supported in OS 1100. The audit data is stored in the System Log File which is a system-high file. Only the SSA can authorize who runs at system high. Anyone who can run at system high implicitly has read access to all audit data. An ACR can be attached to audit files to further restrict access.

OS 1100 provides a configuration parameter that can be set to initiate a system stop if the audit file is ever unavailable or if a security-relevant 1100 Executive activity is temporarily prevented from calling the auditing services. This capability prevents the system from losing audit data without the intervention of an SSA.

Identification and Authentication events are audited by the 1100 Executive and include success or failure as well as the requested mandatory access attributes and the origin of request (e.g., site-id).

Events that introduce objects into the user's address space include: first reference to TIP files in TIP bundles, adding TIP files to a bundle, first access to a tape volume, first access to an 1100 Executive file, and initial common-bank subsystem transition. The individual actions that make up these events are all audited and the mandatory security attributes of the object being accessed are included in these audit entries. All security access failures are audited.

Deletion of cataloged files and removal of TIP files from TIP bundles are audited as object deletion. The mandatory security attributes of the object being deleted are included in these types of audit entries. Failed events of this type are also audited.

The 1100 Executive ensures that all messages sent to the OS 1100 operator and all keyins made by the operator are audited.[1] Also, any modifications to the security database by the SSA are audited. This is not a complete list of audited security-relevant events.

A user who is logged on has a unique user-id. This id is placed in audit records. The double-word starting time (in nanoseconds) of a run or TIP session is used as a unique thread identifier for the run or TIP session. Subsequent audit entries related to the initiated run or TIP session are associated using this thread-id. The audit entries of sessions that are started by a batch/demand TIP-connected program are associated with that program by a sub-thread-id.

The audit reduction tool used on OS 1100 is the Log Analyzer (LA). The Log Analyzer is a comprehensive tool which has an extensive command language with which to analyze the audit file [92].

---

[1] The DCP/OS operator actions are not audited. For this reason the DCP/OS console is not to be operational when the system is running in B1 mode.

Privileges

Generic privileges, not to be confused with Processor Privilege, within the 1100 Executive are secured ERs and Privileges, and each type of generic privilege falls into one of two categories, security-relevant and others. The two types basically use the same mechanism but in a slightly different form. The secured ER is a means to control access to a complete sensitive TCB interface.[1] A privilege is a mechanism to control access to a sensitive Executive subfunction,[2] allow certain security database administration operations, allow certain security policy checks to be bypassed, or allow certain interfaces to function in a different fashion.

The TFM [42] describes all of the privileges, specifies which ones are required to be controlled in a Class B1 system (because they are security relevant or affect system integrity), and explains how to control those privileges that are not initially shipped as controlled.[3]

A specific subset of the security-relevant privileges is called the Trusted Privilege Set (see page 23, "Trusted Subsystems"). If a subsystem has any one of the privileges in the Trusted Privilege Set, that subsystem is defined as a trusted subsystem.[4] Refer to page 23, "Trusted Subsystems" for a description of the handling of a trusted subsystem as opposed to a normal subsystem. A complete list of security-relevant privileges appears on page 63, "Privileges"

Privileges are implemented as a set of control bits, where each bit represents a specific privilege. A unique identifier specified at the time the privilege is defined is used to identify the specific bit in the mask. If a particular bit is set in the privilege mask in the user's security record, then the user is said to have the privilege. Privileges are given to users only by an administrator who has write access to the user-id security record and who also possesses the specific privilege to be given the user. At the time of installation of a subsystem, all privileges allowed the subsystem owner are assigned to the subsystem, and will be in effect whenever it is entered.[5]

---

[1]The ER is the interface.

[2]specifically a subfunction of an ER

[3]Unisys uses the term "enforced" when referring to controlled privileges.

[4]While the evaluation team recognizes that any subsystem with a security-relevant privilege is a trusted subsystem, Unisys has chosen to define trusted subsystem, as stated, to help provide a more concise description of the system security policy.

[5]The (privileged) user cannot restrict or expand the privileges of a subsystem, since this would be a modification of the TCB.

While the mechanisms for privileges and secured ERs are somewhat similar, the specifics for each will be described in the following paragraphs.

Control information for secured ERs is stored in the system ER table accessible only by the Executive. This information consists of a flag that indicates that this ER is secured and the unique ER identifier. The post processing of the ER interrupt is accomplished by an Executive routine (ERSI). Among other things, this processing determines whether the specific ER is to be handled immediately and control returned to the user, whether the user process is to be deactivated, and how the Executive worker that will process the request is to be activated. All of these decisions are based on control information in the system ER table. In addition, if the ERSI determines that the ER is secured, it will call an Executive validation routine (ERVCR) to validate the request. If the user has access to the ER, normal processing continues. If the user cannot perform the ER, a security-abort condition is raised. See page page 35, "Executive Interfaces" for a description of the handling of the security-abort condition.

Privileges operate in much the same way. Executive code that needs to validate a user's right to a privilege calls one of several privilege validation routines (SSPCHK, SSPCHKS or SSGPCHK), based on the specific privilege and the execution environment of the caller. These routines return an indication of whether the user has the privilege or not. It is up to the calling Executive code to determine the action to take based on the response. The B1 security-relevant privileges used in OS 1100 are identified in the following list:[1]

SSADID - Allows absolute device assignment.

SSASCDT - Allows for compartment set creation and change using SIMAN.

SSBAFC - Allows bypass of ACR evaluation for file access and subsystem transition.

SSBKUP - Allows write into read-only files or read from write-only files.

SSBVOLCHK - Allows bypass of volume labeling.

SSBYCL - Allows bypass of clearance level validation.

SSBYCOMP - Allows bypass of compartment validation.

SSBYPASSOWNR - Allows bypass of owner and ACR validations.

SSCCL - Allows bypass of clearance level validation and allows change within the user's range.

---

[1]This classification was verified by the formal team during design analysis and their use was tested.

SSCONSOLE - Allows execution of ER KEYIN$ and privileged ERCOM$ functions.

SSCRCL - Allows clearance level change.

SSCHCOMP - Allows change of compartment set.

SSCONFIGMGR - Allows ER CONFIG$ changes of hardware and software Executive configuration settings.

SSCSPF - Allows modification of special flags.

SSCSU - Allows creation of user security records.

SSDBACK - Allows execution of DBACK$ and DBACK1$ functions of ER MSCON$. Allows FAS backup tape number change in the MFD.

SSDECONTROL - Allows decontrol of compartments.

SSFASSPHNDL - Allows FAS special handling of files.

SSLOGER - Allows creation of log entries.

SSREADEXEC - Allows ER CONFIG$ read of Executive relative address.

SSREMFILOWNR - Allows removal of file owner.

SSSCLDTUPD - Allows change of symbolic representation of clearance level in SCLDT.

SSSMOQUE - Allows execution of ER SMOQUE$ functions.

SSSSCALLANY - Allows call to any subsystem, bypassing all subsystem transition validations.

SSTABCH - Allows SIMAN to change enforcement of ERs and privileges.

SSTIPGETMSG - Allows user to read queued TIP messages using ER QI$CON.

SSTIPSENDMSG - Allows user to send TIP messages to other terminals.

SSMROOC - Allows change of owner in user-id and file security records.

SSRLODCB - Allows reload of common banks using ER BANK$.

SSSTRTZOPT - Allows execution of @START,/Z.

Two sets of trusted privileges are defined:

Trusted Privilege Set A (direct override of security policy)

SSCCL
SSBYCOMP
SSBYPASSOWNR
SSBAFC

Trusted Privilege Set B (override of security policy through special interfaces)

SSSMOQUE
SSADID
SSBVOLCHK
SSDBACK
SSBYCL

The security-relevant ERs are identified in the following list.[1]

ABSAD$ - Allows access to downed main storage.

ACCNT$ - Updates the Summary Account File.

AC$NIT - Initializes TIP online.

AUDIT$ - Provides access to the audit trail.

BANK$ - Reloads common bank or security gate bank after it is installed.

BDSPT$ - Removes a track from available mass storage pool.

CMS$REG - Registers a real-time program as a communications program, with the Executive.

COM$ - Provides communication with system operator.

CONFIG$ - Modifies or inspects system-configurable parameters.

CSI$ - Submits multiple control statements for facilities.

DMABT$ - Notifies Step Control that the Data Management Routine is being aborted.

---

[1]The security relevance of these ERs was verified by the formal team through design analysis and their use was tested.

DM$IO(W) - Performs physical I/O on TIP files.

DNLOD$ - Performs downline load.

DREG$ - A function of ER MSCON$.

EACQ$ - Extent acquire function of ER IO(W)$.

ERCVS$ - Determines whether Step Control has successfully recovered a queue item.

FCREG$ - Permits change to TIP file directory.

FS$UTF - Reads and writes from/to a TIP freespace file.

MCABT$ - Notifies Step Control that MCB processing is being aborted.

MCODE$ - Provides access to microcode loader.

MCT$ - Permits access to the MCT Application Area (10 words writable only with real-time status).

MODPS$ - Modifies privilege state.

MQF$ - Provides recovery and maintenance functions for Step Control queue items.

MSCON$ - Retrieves security information for the cataloged files from the lead item of the MFD.

PB$CON - Permits a batch or demand program to connect to TIP. Does not permit sending of transaction messages.

QI$CON - Permits a batch or demand program to connect to TIP. Permits sending of transaction messages.

REGREP$ - Used in trapping Executive Requests.

REGRTN$ - Used in trapping Executive Requests.

RSI$ - Performs an Executive interface function for the communications control routine.

RT$ - Raises an activity to real-time priority.

RT$INT - Initializes core COMPOOL.

SABORT$ - Aborts security PCT.

SATTCP$ - Compares security attributes.

SCDTA$ - Alters Security Compartment Definition Table.

SCDTL$ - Produces listings of the compartments accessible to the executing user. SIMAN uses the list of the maximum set to edit user-id record updates.

SCLDT$ - Maintains symbolic clearance levels. SIMAN access is controlled by a privilege.

SCOMCNV$ - Converts a security compartment.

SC$QR - Updates Step Control storage.

SC$SR - Recovers a TIP session (Integrated Recovery).

SDEL$ - Deletes user-id and ACR records in SACRD$.

SMOQUE$ - Requests an action to be performed on the symbiont queues.

SPRNT$ - Displays user-id and ACR records from SACRD$.

SREG$ - Creates user and ACR records in SACRD$.

STAB$ - Maintains the list of secured ERs and privileges. This list is used to create the user-id record maintenance screens.

SUMOD$ - Updates the SACRD$ portion of the user-id record and the cataloged file security records in the MFD.

SUVAL$ - Determines user access to other user-id records and cataloged file security records. Checks user against the record's clearance level, compartment set, and Access Control Record. Used by all non- Executive (and some Executive) components of the TCB.

SYSLOG$ - Creates system log entries.

TIP$Q - Creates network, passoff, output, and checkpoint message queue items.

TIP$SM - Opens and closes a TIP session.

TIP$TALK - Controls TIP terminal communications.

TIP$XMIT - Notifies Step Control that an output message has been selected for transmission to CMS 1100 or that output message transmission is complete.

TPFLG$ - Determines operating capabilities of a TIP environment.

TRAPRTN$ - Used in trapping an Executive Request.

TVSLBL$ - Performs tape volume security labeling.

USER$ - Updates user logon profiles in TSS$FILE. (only the SIMAN common bank can execute)

VT$CHG - Updates a main storage copy of a TIP VALTAB record to reflect a change on mass storage.

VT$PUR - Informs OS 1100 concerning update of load control information in a TIP VALTAB file for a particular tra. saction code.

The subsystems in OS 1100 along with the security level and privileges necessary to access them (see page 56, "Mandatory Access Control") are listed as follows:

OS 1100 Component	Security Level (Classification, Compartment Set) B1-Relevant Privileges
MCB Background Run	0-63, ALL SSCCL SSBYCOMP SSBYPASSOWNR
MCB Short/Long	0-63, ALL
MCB	0-63, ALL Trusted Privilege Set A SSTIPSENDMSG SSTIPGETMSG
CMS	0-63, ALL SSCCL SSBYCOMP SSSMOQUE SSADID SSLOGER SSBYPASSOWNR

SIMAN Common Bank	0-63, ALL Trusted Privilege Set A & B USER$ BDI SSCRCL SSCHCOMP User-id executing
SIMAN	needs SSCSU, SSDECONTROL, SSMROOC, SSSCLDTUPD, or SSTABCH for performing various functions.
IRU	0-63, ALL SSCCL SSBYCOMP SSBYPASSOWNR SSSSCALLANY
UDSC	caller's, caller's none
UDS	ADT 0-63, ALL Trusted Privilege Set A
PERCON Background Run	none none
PERCON	0-63, ALL Trusted Privilege Set A SSADID SSLOGER SSCONSOLE SSSMOQUE
TLABEL	caller's, caller's SSBVOLCHK
DPREP	caller's, caller's SSADID
COMUS	caller's, caller's SSBAFC SSCONFIGMGR SSRLODCB

FAS   caller's, caller's
                                                       SSBAFC
                                                       SSCHCOMP
                                                       SSCRCL
                                                       SSCSPF
                                                       SSDBACK
                                                       SSFASSPHNDL
                                                       SSMROOC
                                                       SSREMFILOWNR

The following table describes the common bank subsystem interfaces:

Accessed Entity --->	MCB CB Main	MCB CB Short/ Long	SIMAN	SIMAN Common Bank	UDSC CB	UDS ADT CB	PERCON CB	IRU Run	PERCON Run
Accessing Entity | | V									
User Program	X		X		X				
MCB Run	X								
MCB CB Short/Long	X							X	
IRU Run	X	X			X				
SIMAN				X					
USDC CB	X					X			
PERCON Run (Background Run)						X			
PERCON CB									X
CMS Run	X								

CB = Common Bank

See the TFM [42], Appendix C and Chapter 12, for a discussion of the privileges supported by the 1100 Executive.

Object Reuse

Data is cleared from registers, main storage and mass storage by the TCB before another subject is allowed access to it. For magnetic tapes, the SSA must arrange for degaussing (or some other means external to the TCB) to remove residual data.

On every context switch, all registers are reloaded to the state they had when the process was previously running; the initial register contents are established for a process (activity) by the Executive when the process is first created and include no data from another

process. A few of the activity's registers (that is, the user-writable registers) are initialized to predefined values by the Executive, such as date/time and system identification, and the remaining registers are set to zero. The remaining registers are in the Executive Register set. They can be read but not written by a user process, and contain no security-relevant data.[1] When any new process is spawned (forked) by a user process, the initial register set is identical to that of the spawning process.

Main storage is cleared for all program loads and for all memory requested by an executing program. The system's ability to clear memory is under control of system configuration parameters (one for TIP, and another for batch and demand runs), which must be set for the clearing to take place. Multiple-input TIP transaction programs can leave data residue in common banks and are therefore not permitted on the system.

For mass storage, new blocks are cleared to zeros when a file is created or expanded. This function is also under the control of a system configuration parameter.[2] In addition, an entire mass storage volume can be cleared of data through the use of a utility program (DPREP1100) [75] in the TCB.

---

[1]The formal team verified that this is so.

[2]Unprivileged users have no physical access to files or tape volumes.

This page intentionally left blank.

EVALUATION AS A B1 SYSTEM

Discretionary Access Control

Requirement

> The TCB shall define and control access between named users and named objects (e.g., files and programs) in the ADP system. The enforcement mechanism (e.g., self/group/public controls, access control lists) shall allow users to specify and control sharing of those objects by named individuals, or defined groups of individuals, or by both, and shall provide controls to limit propagation of access rights. The discretionary access control mechanism shall, either by explicit user action or by default, provide that objects are protected from unauthorized access. These access controls shall be capable of including or excluding access to the granularity of a single user. Access permission to an object by users not already possessing access permission shall only be assigned by authorized users.

Applicable Features

OS 1100 enforces a discretionary access control policy over the protected objects of the system (files, tape volumes, and subsystems).[1] The objects are protected by a policy that allows them to be private (access by owner only), public (access to all), or controlled by an access control list. Unisys refers to the access control lists as Access Control Records (ACRs). ACRs provide the capability of including or excluding access to the granularity of a single user. Only an object's owner or the SSA may change the object's access list. Types of access defined for files are read, write, execute and delete. Types defined for tapes are read and write. Only execute access is defined for subsystems and it is interpreted to mean 'enter' access. A newly created object has a default access of private. Access to files and tape volumes is controlled by an 'access list' field in their security records while the access list field for a subsystem is in the owner's user-id security record. An ACR is used to control who can start a batch run for another user-id.

Conclusion

OS 1100 satisfies the B1 Discretionary Access Control requirement.

---

[1]Control is also provided via the read-only, write-only, and exclusive-use mechanisms. However these are not used to enforce the OS 1100 security policy.

Object Reuse

Requirement

> All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation, or reallocation to a subject from the TCB's pool of unused storage objects. No information, including encrypted representations of information, produced by a prior subject's actions is to be available to any subject that obtains access to an object that has been released back to the system.

Applicable Features

Main storage and mass storage clearing by the TCB are available as site-selectable features (see page 72, "Object Reuse".) With the features selected, any such storage area is cleared to zeroes (on allocation) before it is made available to a user after it has been released from previous use. There is no TCB feature for the clearing of magnetic tapes between use, so degaussing must be used to clear tapes.

Conclusion

OS 1100 satisfies the B1 Object Reuse requirement.

Labels

Requirement

> Sensitivity labels associated with each subject and storage object under its control (e.g., process, file, segment, device) shall be maintained by the TCB. These labels shall be used as the basis for mandatory access control decisions. In order to import non-labeled data, the TCB shall request and receive from an authorized user the security level of the data, and all such actions shall be auditable by the TCB.

Applicable Features

All subjects and objects under the control of the 1100 Executive are labeled with a hierarchical classification level (a clearance level) and a non-hierarchical set of categories (the compartment set). Subjects also have a clearance level range. The TCB uses these labels to enforce the OS 1100's mandatory access control policy.

Imported, unlabeled data may be introduced to OS 1100 via removable disk packs and tape volumes. Unlabeled files on removable disk packs are disabled when the pack is registered with the 1100 Executive. Specifically, the 'security disabled' flag (see page 58, "Access to Files") is set in the MFD entry for each unlabeled file. A non-privileged user will not be allowed to assign a file that has this flag set. The SSA must use an 1100

Executive interface to manually set the security attributes of the file and then clear the security disable flag. Note that the 1100 Executive will not prevent the SSA from clearing this flag without setting the security attributes.[1]

An unlabeled tape can only be read by a privileged user.

Conclusion

OS 1100 satisfies the B1 Labels requirement.

Label Integrity

Requirement

> Sensitivity labels shall accurately represent security levels of the specific subjects or objects with which they are associated. When exported by the TCB, sensitivity labels shall accurately and unambiguously represent the internal labels and shall be associated with the information being exported.

Applicable Features

OS 1100 maintains sensitivity labels, consisting of a clearance level and compartment set, for all subjects and objects under the control of the 1100 Executive. Labels for users and subsystems are maintained in the security database (specifically the system file SYS$*SACRD$) (see page 29, "Security Database").

Labels for files are maintained in the MFD entries for those files. Tape volumes are labeled with ISO standard labels which include the security level in VOL3 of the volume header, and also in the MFD entry for that file if the tape volume is cataloged. If the security level in the MFD entry for a cataloged tape is changed (via an ER interface) without also changing the tape header, access to the tape will be denied on subsequent attempts to assign the cataloged tape file. Access, in this case, is only allowed to a privileged user.

Since all these labels can only be manipulated via TCB interfaces, they are guaranteed to be protected from unauthorized modification.

Exportation of information is handled by trusted 1100 Executive activities which ensure that labels are correctly associated with the exported information. Specifically, the only way to export information is on tape volumes, removable disk packs, or print files. Tape volumes are labeled through an ER interface, or automatically by the 1100 Executive on the first write to the volume. Removable disk packs contain the MFD entries for each file on that pack. These MFD entries are only manipulated by the 1100 Executive. Print files

---

[1]Clearing the flag, setting the attributes, and assigning an unlabeled tape are audited.

are labeled by the printer control software PERCON, or by the communications software CMS 1100.

Conclusion

OS 1100 satisfies the B1 Label Integrity requirement.

Exportation of Labeled Information

Requirement

> The TCB shall designate each communication channel and I/O device as either single-level or multilevel. Any change in this designation shall be done manually and shall be auditable by the TCB. The TCB shall maintain and be able to audit any change in the current security level or levels associated with a communication channel or I/O device.

Applicable Features

The multilevel devices in OS 1100 are mass storage devices and printers. Tape drives and terminals are single-level devices that are serially reusable at different levels. These designations are not dynamically changeable by the TCB (i.e., they require a system generation).

The channels connecting these devices to the system (either directly to the central complex or to a front-end processor (FEP)) are considered to be the same level as the devices that they connect, since devices cannot be mixed on these channels. The exception to this is the channel from the host to the FEP. This channel is a multilevel communication channel. Again these designations are not dynamically changeable by the TCB.

Conclusion

OS 1100 satisfies the B1 Exportation of Labeled Information re irement.

Exportation to Multilevel Devices

Requirement

> When the TCB exports an object to a multilevel I/O device, the sensitivity label associated with that object shall also be exported and shall reside on the same physical medium as the exported information and shall be in the same form (i.e., machine-readable or human-readable form). When the TCB exports or imports an object over a multilevel communication channel, the

protocol used on that channel shall provide for the unambiguous pairing between the sensitivity labels and the associated information that is sent or received.

Applicable Features

The multilevel devices in OS 1100 which allow the exportation of data are removable disk packs and printers. In addition, backup tapes are multilevel objects. The I/O channel between the host and the FEP is a multilevel channel.

Disk packs consist of files, and the 1100 Executive guarantees that, for removable packs, the MFD entries for files are maintained on the same pack as the files themselves A pointer to the pack's MFD is maintained in the disk volume header (VOL1).

Printers directly connected to the host are assigned to the controlling software (PERCON) as if they were files. As such the printer and the associated channel can be thought of as being at the same security level as the controlling run. In a similar manner the communications software (CMS 1100 and TELCON) will maintain the connection between the FEP and an FEP-connected printer. As such this connection can be thought to be at the level of the communications software (i.e., system high). However, since these products must handle files at different security levels, they must associate a label with each print file. CMS 1100 or PERCON receive the internal security label for a file when that file is requested from the print queue (ER SMOQUE$). This internal label is converted to its symbolic form through another Executive interface. CMS 1100 and PERCON associate that symbolic label with the file by embedding the label into the transmitted file text for printing.

There are two forms of backup tapes. File backup tapes contain standard backup copies of all files. These tapes are maintained by the trusted utility FAS. FAS records the security label of each file on the tape with the file itself, and accurately restores that label when the file is restored. These backup tapes are labeled system high.

Print queue backup tapes are created by the 1100 Executive to save and restore print files that are on system print queues. The SV and SR operator keyins are used to perform these operations. The 1100 Executive ensures that the security label is written to the tape with the file. The files are restored as PRIVATE files owned by the 1100 Executive with the correct mandatory attributes restored from the security label on the tape. These backup tapes are labeled system high.

The I/O channel connecting the host with the FEP is used by the trusted module CMS executing on the host to communicate with the trusted TELCON software executing on the FEP. This single channel is used to pass all messages to (from) the host from (to) all terminals connected to that FEP. CMS and TELCON use a unique session identifier to identify each active session. Each message passed across the channel contains that unique identifier to insure the proper message routing. The Executive maintains the security attributes associated with each of these sessions.

Final Evaluation Report UNISYS OS 1100
Evaluation as a B1 System

Conclusion

OS 1100 satisfies the B1 Exportation to Multilevel Devices requirement.

Exportation to Single-Level Devices

Requirement

> Single-level I/O devices and single-level communication channels are not
> required to maintain the sensitivity labels of the information they process.
> However, the TCB shall include a mechanism by which the TCB and an
> authorized user reliably communicate to designate the single security level
> of information imported or exported via single-level communication
> channels or I/O devices.

Applicable Features

The OS 1100 supports two types of single-level devices: terminals and tape drives. The
level of a session associated with a particular terminal is set during the sign-on validation
routines and this level also applies to the communication channel from the front-end
processor to the terminal.

The level of a tape device is the level of the tape that is mounted on that device. At the
first I/O to a newly mounted or remounted tape, the Executive reads the tape label and
establishes the device level based on the level from the label. In those cases where the
tape does not have a label (can only be referenced by a privileged user) or does not have
a security level in the label (a prelabeled blank tape), the device has the level of the user
run that is using that tape.

Conclusion

OS 1100 satisfies the B1 Exportation to Single-Level Devices requirement.

Labeling Human-Readable Output

Requirement

> The ADP system administrator shall be able to specify the printable label
> names associated with exported sensitivity labels. The TCB shall mark the
> beginning and end of all human-readable, paged, hardcopy output (e.g., line
> printer output) with human-readable sensitivity labels that properly*
> represent the sensitivity of the output. The TCB shall, by default, mark the
> top and bottom of each page of human-readable, paged, hardcopy output
> (e.g., line printer output) with human-readable sensitivity labels that
> properly* represent the overall sensitivity of the output or that properly*

represent the sensitivity of the information on the page. The TCB shall, by default and in an appropriate manner, mark other forms of human-readable output (e.g., maps, graphics) with human-readable sensitivity labels that properly* represent the sensitivity of the output. Any override of these marking defaults shall be auditable by the TCB.

Applicable Features

The SSA uses a privileged interface of the site management software, SIMAN, to associate a symbolic name with each clearance level and compartment valid on the system.

For print output, all pages of one print file are the same sensitivity level, which is marked on banner and trailer pages by PERCON or CMS 1100. By default, the top and bottom of every page are also marked with the sensitivity label. Printed labels are one to four lines long, using up to 225 characters. If the number of compartment indicators makes a label too long, the system labels the affected printout with the appropriate clearance level, the compartment set ALL and the phrase "(The security label is too long to be printed)".[1] An audit log entry stating that fact is created. This mechanism supports the proper physical handling of the printout.

Users may disable top and bottom page labeling. This act will cause a log entry to be written. In addition, a log entry is created, and top and bottom page labeling automatically disabled, if the TCB cannot ensure that every page is properly labeled. This situation might arise on some printers if a format change occurs that leaves no room for the label or if a font change is specified that would destroy the label appearance.

PERCON and CMS 1100 print sequence numbers on each banner and trailer page to help operators detect spoofing attempts where users generate bogus banners and trailers. The sequence number will be reset with some random value every time PERCON or CMS 1100 starts processing a specific print queue, and will be incremented by one for each printout in the queue until the queue is depleted.

Conclusion

OS 1100 satisfies the B1 Labeling Human-Readable Output requirement.

---

[1] See page 10-3 of the TFM [42]. Also, rfc 2140 states that if a site uses a paper width less than 8.5 inches, or a font that prints at less than 10 cpi, the security label (on the top and bottom of each page) will be truncated. Since this situation cannot always be detected, and thus no log entry generated to indicate this situation, the TFM contains warnings about the use of paper and fonts which will not allow the full label to be printed.

Mandatory Access Control

Requirement

> The TCB shall enforce a mandatory access control policy over all subjects
> and storage objects under its control (e.g., processes, files, segments,
> devices). These subjects and objects shall be assigned sensitivity labels that
> are a combination of hierarchical classification levels and non-hierarchical
> categories, and the labels shall be used as the basis for mandatory access
> control decisions. The TCB shall be able to support two or more such
> security levels. The following requirements shall hold for all accesses
> between subjects and objects controlled by the TCB: A subject can read an
> object only if the hierarchical classification in the subject's security level is
> greater than or equal to the hierarchical classification in the object's security
> level and the non-hierarchical categories in the subject's security level
> include all the non-hierarchical categories in the object's security level. A
> subject can write an object only if the hierarchical classification in the
> subject's security level is less than or equal to the hierarchical classification
> in the object's security level and all the non-hierarchical categories in the
> subject's security level are included in the non-hierarchical categories in the
> object's security level. Identification and authentication data shall be used by
> the TCB to authenticate the user's identity and to ensure that the security
> level and authorization of subjects external to the TCB that may be created
> to act on the behalf of the individual user are dominated by the clearance
> and authorization of that user.

Applicable Features

OS 1100 enforces a mandatory access control policy over the defined subjects in the
system (subsystems) and the system's objects (files, tape volumes, and subsystems). A
clearance level, consisting of 64 possible security levels and a compartment set, consisting
of a maximum of 30 possible compartments is associated with each subject and object [4].

A subject can read data from an object only if the security level of the subject dominates
that of the object. A subject can write to an object only if the security level of the subject
equals the level of the object. See page 83, "Identification and Authentication" concerning
subsystem transition.

Conclusion

OS 1100 satisfies the B1 Mandatory Access Control requirement.

Identification and Authentication

Requirement

The TC⸱ shall require users to identify themselves to it before beginning to perform any other actions that the TCB is expected to mediate. Furthermore, the TCB shall maintain authentication data that includes information for verifying the identity of individual users (e.g., passwords) as well as information for determining the clearance and authorizations of individual users. This data shall be used by the TCB to authenticate the user's identity and to ensure that the security level and authorizations of subjects external to the TCB that may be created to act on behalf of the individual user are dominated by the clearance and authorization of that user. The TCB shall protect authentication data so that it cannot be accessed by any unauthorized user. The TCB shall be able to enforce individual accountability by providing the capability to uniquely identify each individual ADP system user. The TCB shall also provide the capability of associating this identity with all auditable actions taken by that individual.

Applicable Features

All users (demand, batch, and TIP) must identify and authenticate themselves to 1100 Executive by presenting a valid user-id and password combination before performing any subject-specific actions.[1] The team has examined those commands that can be entered prior to identification and authentication and has determined that no security breach will occur. Such commands include those that set terminal characteristics and those that establish a connection with the communications software. After identification and authentication, and before (or after) a run is initiated by the user, operator console commands may be entered from a terminal, subject to the user's privilege restrictions.

Userids and passwords are stored in the file TSS$FILE which also contains information such as last login, number of login attempts for this user-id, valid project numbers and valid account numbers. Every record in TSS$FILE has a corresponding user security record in the file SACRD$ which identifies the user's clearance level range, maximum compartment set, privileges, etc. The file TSS$FILE can only be accessed through SIMAN, and then only if the user operating SIMAN has the appropriate privileges.

---

[1] Although operators are users, the NCSC found it acetable to allow a paper operators log to satisfy this requirement for operators only. Operator commands permitted to terminal users via the @@CONS command and submission of batch runs via the @START command are identified by the user signed onto the terminal session.

SIMAN ensures that all user-ids are unique over the lifetime of the system. When a user logs on, a thread-id is established which is associated with all actions taken by that user. Audit records note this association, and the audit reduction tools can translate back from thread-id to user-id.

Configuration parameters let the SSA set the maximum number of invalid logon attempts that will be permitted, determine what happens when this limit is reached, enforce minimum password lengths (the lowest value is 1), and enforce password aging.

The 1100 Executive ensures that subsystems operating on behalf of a user do not make a transition to any untrusted subsystem (i.e., a subsystem with no privileges from the Trusted Privilege Set) that is operating at a level beyond that authorized for the specific user.

Conclusion

OS 1100 satisfies the B1 Identification and Authentication requirement.

Audit

Requirement

> The TCB shall be able to create, maintain, and protect from modification or unauthorized access or destruction an audit trail of accesses to the objects it protects. The audit data shall be protected by the TCB so that read access to it is limited to those who are authorized for audit data. The TCB shall be able to record the following types of events: use of identification and authentication mechanisms, introduction of objects into a user's address space (e.g., file open, program initiation), deletion of objects, actions taken by computer operators and system administrators and/or system security officers, and other security relevant events. The TCB shall also be able to audit any override of human-readable output markings. For each recorded event, the audit record shall identify date and time of the event, user, type of event, and success or failure of the event. For identification/authentication events the origin of request (e.g., terminal ID) shall be included in the audit record. For events that introduce an object into a user's address space and for object deletion events the audit record shall include the name of the object and the object's security level. The ADP system administrator shall be able to selectively audit the actions of any one or more users based on individual identity and/or object security level.

Applicable Features

Unisys provides a comprehensive and thorough audit system on OS 1100 . The interface into the operating system used is the EXEC LOG, which is a file owned by the operating system and accessible only to system-high users. Unisys has increased the number and

types of events that are passed to this log to fulfill the audit requirement of the TCSEC. All security-relevant events are always audited The audit process captures all command execution, job creation and start of ERs, use of privilege, generation of output, object access, and both actual security policy violations and attempts.

Unisys provides an audit reduction tool which appears to meet the needs of the real-world operational System Security Administrator. The tool (. processor) [92] provides a means by which the SSA may track individual users and their activities or specific activities, as well as access to objects based on their security level.

Conclusion

OS 1100 satisfies the B1 Audit requirement.

System Architecture

Requirement

> The TCB shall maintain a domain for its own execution that protects it from external interference or tampering (e.g., by modification of its code or data structures). Resources controlled by the TCB may be a defined subset of the subjects and objects in the ADP system. The TCB shall maintain process isolation through the provision of distinct address spaces under its control. The TCB shall isolate the resources to be protected so that they are subject to the access control and auditing requirements.

Applicable Features

The main processor hardware used with OS 1100 [37] provides several features for use in supporting the TCB isolation and protection requirements. Memory references are descriptor-based, with the descriptor containing access rights information in the form of a lock and two access permission fields (see page 10, "Use of Keys and Locks"): one for special access, and the other for general access. The Executive is protected and subsystems are isolated via the key and lock mechanism in the hardware.[1] Subsystems are constructed using the key and lock mechanism along with gate banks.

The key used with the lock is part of a process's state. The TCB assigns keys and locks such that the TCB's memory and data space are protected from modification by a user

---

[1] For example, Executive banks have a lock whose ring value is zero and whose domain values are mutually exclusive with user domains. No user activities have a ring value less than zero or a domain value equal to an Executive domain, so there will never be a match of key and lock between the user and the Executive. Thus General Access Permission is in effect. The bits defining General Access Permission are always set to prohibit access to the Executive.

process. The manipulation of the keys and locks can be done only when the IP is executing in a privileged[1] state, which is reserved for the execution of the Executive portion of the TCB. The descriptor-based addressing scheme facilitates the sharing of TCB addressability among user processes, while isolating the private space of each user process from that of any other user process. Two descriptor tables which are shared among users point into the Executive and other TCB components, as well as to the site-installed common bank subsystems. A third descriptor table, one per run, is used to address a run's private data space (see page 9, "Instruction Processor (IP) and Main Storage").

TCB mediation is provided for accesses between subjects and objects (see page 53, "Access Control"), as well as for the auditing of such accesses (see page 63, "Audit"). A special hardware mechanism, gates, (see page 9, "Instruction Processor (IP) and Main Storage") is provided for the controlled entry into subsystems that may either be trusted, or may share data from several users.

Conclusion

OS 1100 satisfies the B1 System Architecture requirement.

System Integrity

Requirement

> Hardware and/or software features shall be provided that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TCB.

Applicable Features

Unisys has a set of maintenance diagnostics that include both online and offline modes. The following components comprise the major portion of the Diagnostic Verification Routines for the OS 1100: Online Confidence Instruction Processor (OLCIP); Online Main Storage (OLMSU); Mass Storage Test (MST); Central Complex Diagnostic Test Software (CCDTS); Peripheral Test Sequencer (PTS); and Diagnostic Offline Controller (DOC). These routines and programs can be used to verify the correct operation of all system hardware and firmware. The use of these diagnostic test tools assures the SSA that the hardware is operating properly and meets the specification of the applicable model : 1100/90 and System 11 [37], and 2200/200 [38]. The diagnostics identified allow the testing of all hardware and firmware components to be performed in ι predictable and positive manner.

---

[1]Although the processor supports four processor privilege levels, only two are used: one for the 1100 Executive, and the other for all other processes.

Conclusion

OS 1100 satisfies the B1 System Integrity requirement.

<u>Security Testing</u>

Requirement

> The security mechanisms of the ADP system shall be tested and found to work as claimed in the system documentation. A team of individuals who thoroughly understand the specific implementation of the TCB shall subject its design documentation, source code, and object code to thorough analysis and testing. Their objectives shall be: to uncover all design and implementation flaws that would permit a subject external to the TCB to read, change, or delete data normally denied under the mandatory or discretionary security policy enforced by the TCB; as well as to assure that no subject (without authorization to do so) is able to cause the TCB to enter a state such that it is unable to respond to communications initiated by other users. All discovered flaws shall be removed or neutralized and the TCB retested to demonstrate that they have been eliminated and that new flaws have not been introduced.

Applicable Features

The formal evaluation testing is described:

The purpose of the testing effort was to gain the assurance that the TCB design, as evaluated earlier by the team, was implemented correctly. To do this, the team met with the TCB developers and identified the types of user interfaces into the TCB (see Appendix G, Section G.1). Following this activity, the team identified, with Unisys, every user interface into the TCB (see Appendix G, Section G.2). Using this interface list, as a guide for coverage, the team outlined an acceptable test plan which Unisys ultimately developed into a suite of tests that provided acceptable test coverage (see Appendix G, Section G.4). Finally, to provide a convincing model equivalency argument for the security relevant system components of the three distinct models under evaluation (System 11, the 2200/200 and the 1100/90), Unisys ran their entire suite of test on all three models (see Appendix G, Section G.3). The evaluation team selected individual tests, at random, and compared the results from all three evaluated models.

Security testing, started with the team verifying model hardware configurations. Since the System 11 and the 2200/220 were dedicated systems throughout the testing period, this hardware verification was only required once. The 1100/90 was not a dedicated system during the entire testing session, so, hardware verification occurred at the beginning of every testing session on the this model.

Once the hardware configuration was verified, the team built the OS 1100 Executive, which is the core of the TCB. Next, the team installed all of the remaining TCB products (see Appendix G, Section G.5). This TCB installation process was completed on the 2200/200. Unisys then performed the system generation process on the 1100/90 and the System 11. The evaluation team compared the results of all three system generations to ensure that the same TCB resided on all three models.

The evaluation team ran one complete suite of tests provided by Unisys. Individual tests were run on one of the three evaluated models until the entire test suite had been run. Since all models had equivalent TCBs, this parallel testing expedited the test process. Once the entire Unisys test suite had been completed, the evaluation team then selected tests to run on each model and compared the results with those previously run by Unisys for equivalency. Finally, the evaluation team selected tests run on a particular model and processed these tests on all models and checked results for equivalency (see Appendix G, Section 7).

The team performed code reviews of selected tests, specifically those that: attempted all possible machine instructions, attempted illegal ERs, demonstrated object reuse in memory as well as mass storage, and those tests which satisfied the testing requirement for multiple ERs by using a 'gray-box' approach. This gray-box approach was employed by Unisys to reduce test cases. In several areas, common ERs with different ER indexes were tested by eliminating redundant tests when a collection of interfaces all use a common security-relevant reference routine. An example of this is the symbiont ERs, which all cause an ER to the IO routine, which is the same routine used when user programs perform ER IOW$. Therefore, once all access rules for ER IOW$ have been verified, a subset of symbiont ERs, showing that appropriate access checks are occurring, is sufficient to cover all the symbiont ERs.

The TCB interfaces where the 'gray-box' test reduction has been employed are:

- Symbiont ERs for alternate files (READA$, PRNTA$, PNCHA$, SYMB$, etc.)
- Program File Package ERs (PFI$, PFS$, etc.)
- ASCII version of ERs (AREADA$, APRINT$, etc.)
   ECL statements @ADD, @BRKPT and @SYM
- Transparent Control Statements (@@ASG, @@FREE, etc)

Problems Uncovered During OS 1100 Testing

There were no real problems encountered during the final testing session for this product. Numerous previous testing sessions conducted by Unisys and the evaluation team had resolved nearly all issues. Some of the test suites, which were the earliest ones written, were difficult to evaluate because they attempt to test an entire security feature. Later tests are more concise, better written, test one user TCB interface and, therfore, much easier to evaluate. All team questions, which arose trough the test evaluation process, were answered satisfactorily by Unisys.

Lastly, the team performed the sixteen team test described in Appendix E. All test performed satifactorly.

Design Specification and Verification

Requirement

> A formal or informal model of the security policy supported by the TCB shall be maintained over the life cycle of the ADP system and demonstrated to be consistent with its axioms.

Applicable Features

Unisys has supplied to the team a document |4| that provides a description of their philosophy of protection and how it is translated to the TCB. The document also states the informal security model for OS 1100 and enumerates the TCB by module, showing how the modules support the model and showing the interfaces between TCB modules.

Conclusion

OS 1100 satisfies the B1 Design Specification and Verification requirement.

Security Features User's Guide

Requirement

> A single summary, chapter, or manual in user documentation shall describe the protection mechanisms provided by the TCB, guidelines on their use. and how they interact with one another.

Applicable Features

Unisys has produced an acceptable SFUG |91|. The SFUG introduces the concept of Security Option Levels (SECOPTS) in which Unisys is marketing OS 1100 security features. Security is being provided to customers as Security Options, or SECOPTs. Each SECOPT level depends upon the installation of the previous level, so that security features are added until a customer reaches "SECOPT 3 Trusted Environment", the B1-rated system.

A description of the SFUG document follows.

In the Introduction, the purpose of this document and the intended audience are accurately identified.

Chapter One is an introduction to system security. Problems associated with unsecure systems are reviewed, followed by a brief discussion of OS 1100 system security. Subjects and objects are defined as well as MAC and DAC.

Chapter Two presents a high-level overview of the OS 1100 security system and describes specific features. The concept of purchasing security features in additive packages is presented. Five packages build in a cumulative manner to produce the B1 system (SECOPT 3 Trusted Environment).

Chapter Three discusses system access. Users are instructed in the creation and use of passwords, clearance levels, compartments, project-ids and account numbers in reference to system logon and run initiation. Examples are given.

Chapter Four examines mass storage file security. MAC rules and their effect on the user and the appropriate use of DAC are discussed with examples. As part of the DAC discussion, the use of ACRs and the read only/write only mechanism are explained. Examples are given.

Chapter Five discusses printout labeling. Page labeling, how to override page labeling and the auditing of labeling override are explained, as is transporting labeled printouts and simple spoofing prevention.

Chapter Six describes the protection mechanisms for tape volumes. Labelled tapes are defined and MAC and DAC validation for tapes are explained. Examples are given.

Chapter Seven examines protecting common banks. Trusted and untrusted subsystems, home subsytems, and subsystem transition are discussed. Privileges associated with common bank subsystems are given and guidelines for users writing common bank subsystems are given briefly. Useful tables and examples are given.

Chapter Eight briefly discusses protecting labeled and non-labeled removable objects which includes removable disk packs and non-labeled tape volumes.

Chapter Nine describes the use of TIP security features. This includes TIP session control, TIP message security, TIP file security, and a mention of developing TIP programs.

Chapter Ten discusses auditing by listing all events that are logged, with explanatory text.

A glossary of terms is provided.

The SFUG document is well written. Sentence structure is short and simple and the vocabulary makes this document easily readable by a large segment of the user population. Also the SFUG is sufficiently detailed to be quite readable without referenced material, but not so voluminous that it would deter the average user from reading it.

The Unisys SFUG provides specific references (document title, chapter, and section) to other documents for detailed information. This approach is considered acceptable by the TCSEC and the evaluator community.

Conclusion

OS 1100 satisfies the B1 Security Features User's Guide requirement.

Trusted Facility Manual

Requirement

> A manual addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility. The procedures for examining and maintaining the audit files as well as the detailed audit record structure for each type of audit event shall be given. The manual shall describe the operator and administrator functions related to security, to include changing the security characteristics of a user. It shall provide guidelines on the consistent and effective use of the protection features of the system, how they interact, how to securely generate a new TCB, and facility procedures, warnings, and privileges that need to be controlled in order to operate the facility in a secure manner.

Applicable Features

The Trusted Facility Manual provided by Unisys is entitled "Planning and Administration Reference Manual" [42]. This manual is in a series of manuals identified as "UNISYS OS 1100 Security System." The manual has sixteen chapters and nine appendices which are identified below.

Chapter One introduces the concepts and importance of computer system security. A general overview of the security system is provided which includes a discussion of password security, preventing unauthorized access, MAC, DAC, clearance levels, compartment sets, labeling, subjects, objects, data bases for controlling system security and the general responsibilities of a System Security Administrator.

Chapter Two identifies configuration parameters which are necessary to control in a B1 environment. Each parameter is described briefly. This chapter is designed as a concise aid to be used when performing a system generation.

Chapter Three describes how to create a B1 (SECOPT 3 Trusted) environment. Appropriate system parameter settings are identified. Appropriate settings for the TIP environment are identified. All software included in the TCB is identified and appropriate control parameter settings are identified. This chapter reflects an effort by Unisys to

comply with an evaluation team request that a description of everything required to establish a B1 environment be presented in one location in the TFM, and that this information be presented early in the document.

Chapter Four presents an overview of the B1 system initialization process during an initial or recovery boot.

Chapter Five discusses creating and administering the mechanisms which enforce mandatory access controls. Topics include clearance levels with their associated symbolic names and the system compartment set. The concept of the Security Mandatory Definition Table (SMDT) is presented. The table exists in the system file SYS$*SMDT$ and instructions on how to create and manage this file are presented. As part of this discussion, the methodology for creating, renaming, modifying, decontrolling, and reusing compartments is described.

Chapter Six discusses creating and managing the database which identifies all valid clearance levels and compartment sets. The five files of the security database are identified and explained. The set of all valid user-ids, passwords, account numbers, project-ids, console capability, application control information, and password expirations are identified in one file. All valid compartment and clearance definitions are identified in a second file, while a third identifies all valid file security attributes. A fourth file identifies account numbers and associated user-ids and resource maximums. Finally, a fifth file identifies user profiles and access control records.

Chapter Seven describes the methods to install and delete users and how to specify run modes and security records. Specific discussion includes creating subadministrators, installing and deleting user-ids, creating user-id security records, specifying attributes in the user-id security record, granting special privilege, specifying privilege to change run clearance levels (highlighted as an individual section), and modifying security records.

Chapter Eight identifies methods of operation which help ensure the security of the system. Features described include hacker frustration, password control, project-id control, controlling the security officer user-id and password, and starting runstreams at various security levels and with various system options.

Chapter Nine describes file security administration. Ownership and access are described in general terms. Creating files, assigning files, and MAC are discussed in relation to specific system privileges. Symbiont file attributes are discussed, as are administrative controls for system files.

Chapter Ten discusses appropriate printout labeling and system parameters which must be set to configure the printout labeling feature.

Chapter Eleven describes tape volume operational security. Configuration parameters which affect tape volume security are identified. A guideline for tape handling, a list of

system privileges relating to tape volume security, and brief discussions of DAC for tapes and migrating to a secure system are provided.

Chapter Twelve discusses trusted and untrusted common bank subsystems. The privileges of trusted common bank subsytems are identified. Configuring common bank protection and assigning security attributes to common bank subsystems also are described. The TFM states that installing a new trusted subsystem invalidates the B1 rating for that particular system.

Chapter Thirteen discusses handling and transporting labeled and non-labeled imported and removable objects.

Chapter Fourteen provides information about the TIP environment. A general discussion about TIP environment protection is followed by descriptions of TIP installation parameters, TIP log entries, error notification, IRU usage, application development, and cautions.

Chapter Fifteen discusses auditing security-relevant events. References point to appropriate manuals for additional information. A table of log entry types with brief descriptions is provided.

Chapter Sixteen provides guidelines for the hardware maintenance and diagnostic capabilities of OS 1100 systems having Class B1 configurations.

Nine appendices are included in the TFM. Appendix A lists all ERs which are security relevant and provides a detailed discussion of each. Appendix B briefly discusses those ERs which are secured on the release tape as well as those which must be secured at system initialization. Appendix C describes all privileges. Appendix D identifies all logged security-related error message types. Appendix E presents the PCT Security Buffer format. Appendix F describes the SMDT format. Appendix G discusses the formats of the files in the security database. Appendix H describes the physical label on tapes. Appendix I describes all operator messages that appear during system initialization.

The TFM also provides a glossary of terms and an index.

Conclusion

OS 1100 satisfies the B1 Trusted Facility Manual requirement.

Final Evaluation Report UNISYS OS 1100
Evaluation as a B1 System

Test Documentation

Requirement

The system developer shall provide to the evaluators a document that describes the test plan, test procedures that show how the security mechanisms were tested, and results of the security mechanisms' functional testing.

Applicable Features

Review of Vendor Test Plan

Unisys has prepared a B1 Security Test Plan Revision 11, dated 29 August 1989 which includes a comprehensive mapping of the tests to the Criteria and to all external interfaces of the TCB [33]. Adequate test coverage for the security-related features of the system is included. An adequate test has been provided for each interface. The current test documentation describes the philosphy of testing and each individual test description describes the purpose of the test, the TCB interfaces tested, the resons for choosing the specific testing technique, test procedures describing each step of the test, and the expected results. Also, a script is provided, to run each test, which easily can be followed.

To achieve the goal of satisfying the B1 requirements, Unisys has provided the following three categories of tests descriptions:

1. The B1 system is built and installed according to the TFM (see Appendix G, Section G.5)

2. All security-relevant user interfaces to the TCB are tested; i.e., Executive Control Language (ECL), Executive Requests (ER), etc (see Appendix G, Section G.4).

3. All B1 TCSEC requirements which are not direct user interfaces are tested; i.e., Object Reuse, Audit, etc (see Appendix G, Section G.6)

The team found this documentation to be minimally acceptable. Problems still exist that caused significant procedural work during testing. Specifically, there is not always a clear mapping between the purpose, procedures and expected results. Determining whether or not some test passed is a laborious task.

Conclusion

OS 1100 satisfies the B1 Test Documentation requirement.

Design Documentation

Requirement

> Documentation shall be available that provides a description of the manufacturer's philosophy of protection and an explanation of how this philosophy is translated into the TCB. If the TCB is composed of distinct modules, the interfaces between these modules shall be described. An informal or formal description of the security policy model enforced by the TCB shall be available and an explanation provided to show that it is sufficient to enforce the security policy. The specific TCB protection mechanisms shall be identified and an explanation given to show that they satisfy the model.

Applicable Features

Unisys has prepared a document that provides a description of their philosophy of protection and how it is translated to the TCB. The Design Document [4] is reasonably well written and complete. It has five chapters and one appendix.

The preface describes the purpose of the document and identifies all the TCB components.

Chapter one details the philosophy of protection for OS 1100. The following concepts, from a Unisys perspective, are introduced: system security controls, subjects and objects, identification and authentication, access controls, privilege, hardware protection mechanisms, and multiprocessing.

The second chapter discusses the supported configurations for a B1-rated system. Hardware not available on a B1 configuration is identified with a brief description of associated security relevant-problems. Restricted software is also identified and reasons for exclusion on a B1 system are briefly discussed. There is a brief discussion of the limited communications capabilities on the evaluated system as well as a list of communications software not allowed on the evaluated system. Finally, a list of products is included which were unevaluated because of resource and time limitations.

Chapter three is a discussion of the OS 1100 security policy. The identification of subjects and objects in the system is thouroughly discussed. The explanation of MAC mechanisms includes: clearance level, compartments, and MAC rules. An explanation of DAC discusses three types of access lists and their purpose. Next, the following security-relevant aspects of subsystems are explained: user I&A, home subsystems, common bank subsystems, untrusted subsystems, single- and multilevel subsystems and library subsystems. The chapter ends with a detailed description of the security records in the security data base, which includes user and subsystem, ACR, file, and tape volume records.

Chapter four enumerates the TCB. Each TCB component and the reason for its inclusion in the TCB are explained briefly. Also listed are products not in the TCB, but included in the evaluated system.

Chapter five details the security-relevant design of the TCB modules. This chapter comprises most of the Design Document. The chapter begins with a diagram of the TCB and a detailed discussion of the TCB interfaces which includes: user and operator, intra-TCB, and a schematic of all the common bank interfaces. Next, the security profile for each TCB component outside the Executive is provided. Since each of these TCB components is a pseudo-user, each has a security profile, complete with processor privileges, clearance level, compartment set, and trusted privileges. All of these profile attributes are listed for CMS1100, COMUS, DPREP 1100, FAS, IRU, MCB, PERCON, SIMAN, TLABEL, and UDS. Following this discussion is a detailed discussion of all system software provided on the B1 system.

The Design Document has one appendix entitled "OS 1100 System Concepts." Information provided here includes:

- Activity and task discussions

- Virtual addressing

- Locks and keys

- Designator register

- Gate mechanism

- Home subsystems

- Bank architecture

- Differences in 1100 and 2200 architecture.

Conclusion

OS 1100 satisfies the B1 Design Documentation requirement.

ASSURANCES

Functional Testing

The testing performed by the evaluation team focused on four areas. First the team and Unisys worked closely to develop a test suite which provided tests to ensure the B1 system can be generated correctly by competent individuals who may not have previous Unisys experience except the type of training typically provided in a classroom environment for new customers. Second, Unisys appropriately changed testing emphasis from attempting to exhaustively test security mechanisms to exhaustively identifying all user interfaces into the TCB and testing all the security relevant code for each interface. Unisys provided a comprehensive suite of test which they ran on the System 11, the 2200/200 and the 1100/90 sytem configurations. This suite of tests was run by the evaluation team and results were compared with the results presented by Unisys.

Third, the team augmented the Unisys test suite by focusing on system areas which had been made more secure as a result of this evaluation process and by creating test which focused on a particular feature which was not tested in the Unisys test suite (see Appendix E).

Last, the team asked for demonstrations of certain methods used by Unisys to determine whether the methods meet security requirements (i.e. audit, hardware diagnostics, controls of software releases).

System Generation

Unisys provided test suites which were specific scripts for creating a B1 environment. These test suites identified every step required to build a B1 Exec, install COMUS, and use COMUS to install other required B1 products. The evaluation team was able to perform a complete system generation and is satisfied that a new Unisys customer, with minimal training, has a reasonable degree of assurance that the B1 product can be successfully installed.

Interface Testing

Initially, the Unisys test plans were to verify that the new features required for the B1 system were correctly implemented for all environments, not just the B1 system. These efforts were performed by individual development organizations. These unit tests were executed before the new feature code was integrated into the Executive system. When Formal Evaluation began, the applicable feature tests were selected and only the B1 portions were retained. This effort was to demonstrate that the B1 system provided the required B1 security mechanisms.

Next, Unisys attempted to perform integration testing by creating tests which would exhaustively test one TCSEC requirement. This effort proved to be an unworkable solution to providing the appropriate assurance required.

Further NCSC clarification of security testing led to the identification of all security relevant interfaces into the TCB and the addition of tests for all TCB security-relevant interfaces; e.g., every Executive Request (ER) and Executive Control Language (ECL) command that is security relevant was added to the test plan. Thus, the current test plans have evolved into a product assurance test suite that demonstrate that the evaluated design of OS 1100 has been correctly implemented.

Most test cases use a 'black-box' approach where the test were designed to be exercised from an external interface. However, some security relevant interfaces use the same test to demonstrate correct implementation. This "gray-box" approach was used to reduce test cases only where common ERs with different ER indexes share a common reference routine.

Team Tests

The team tested the absence of debug code in the TCB which could be used to circumvent the system security policy. Specifically, the use of two ERs, not published in user-available documentation, and debug code in the SSP were tested.

Proper control of attempted instruction execution was the focus of three team tests to determine the outcome of attempted execution of illegal ERs, operator keyins, and the outcome of all machine instructions when executed by a user.

DAC on tape was tested to ensure that preassigned tapes had appropriate DAC when used by unprivileged users regardless of which of the two available ERs is used. Also, the team attempted to access the label on tape to determine if the label could be manipulated by a user.

Configurations of specific peripheral devices was the focus of 2 team tests, specifically, console reconfiguration and local printer non- configuration.

Two products tested by the team are COMUS and SIMAN including the establishment of sub-administrators and console privilege inheritance

Many candidate team tests were discussed with Unisys and were encorporated into their test suite.

Demonstrations During Final Testing

The team requested demonstrations of OS 1100 audit capabilities, hardware diagnostics, and software configuration controls.

Audit

Unisys successfully demonstrated that appropriate actions were audited in the system and that the log reduction tool provided sufficient reporting at the appropriate granularity of a single user.

Hardware Diagnostics

Unisys' OS 1100 provided online diagnostics which are the same as typically provided at customer sites. The online diagnostics consist of: online confidence instruction processor test, online main storage confidence test, online mass storage confidence test and online peripheral maintenance tests for disks, tapes and card reader peripherals. The diagnostic tests are sent on tape with every system along with a complete description of how to run the tests. The test programs are menu driven and run without privilege. The tests are capable of making as many passes through the system as the user wishes. A description of maintenance and the diagnostics exists in the TFM. The online diagnostics were run before an evaluation team member and were found to be acceptable.

Configuration Mangement

The team verified the actual release numbers for all TCB components, including the latest patches, called Problem List Entries (PLEs). The team was convinced that Unisys can identify TCB component releases to the accuracy of an individual PLE. These release numbers with associated PLE numbers were used to update Appendix B.

This page intentionally left blank.

EVALUATORS' COMMENTS

This section contains comments and opinions from the evaluation team in the following areas:

- security features provided by the system that are not required by the Criteria.

- the usability of some of the features that satisfy the Criteria requirements.

- the adequacy of the features in satisfying the Criteria requirements.

1. Operator consoles lack an identification and authentication mechanism. This requires a site to rely on procedural control to meet the I&A requirements.

2. The system supports separate operator and security administrator roles. This limits the role of the operator and provides a more secure environment.

3. The Master File Directory entries are visible to all users. The presence of this structure in the system is marginally acceptable.

4. The combination of the use of the SOE (start-of-entry) on a terminal followed by the command $$OPEN provides a trusted-path for logon.

5. Implementation of compartment bits is such that maintaining compartment equivalency between systems is impossible. Even though compartments may be defined on multiple systems in the same sequence, the compartments will not be equal because compartment definitions are based on time-stamps. To implement the same compartment definitions on multiple systems, it is necessary to define a compartment set on one system and copy the definition files to other systems.

6. Unisys has implemented a discretionary access control mechanism that they call Access Control Records (ACRs). ACRs are an implementation of access control lists, a mechanism not explicitly required until B3. However, the implementation of ACRs are static which makes them difficult to use.

7. During the Formal Evaluation phase of the NCSC evaluation, there were multiple instances when it was evident that there was no central configuration control over the entire TCB.

8. All security-relevant events are always audited, not just auditable.

This page intentionally left blank.

EVALUATED HARDWARE COMPONENTS

Evaluated Hardware

Three major system types are evaluated: System 11, 1100/90, and 2200/200. All three system types share a common system architecture (internally known as C-Series) which is documented in [37] and [38]. C-Series architecture machines are often referred to as "Extended Mode Architecture" machines, as C-Series was the first to support the extended mode architecture.[1]

System 11

The System 11 is a small scale system which can have from 1 to 4 instruction processors and from 1,048K to 4,194K 36-bit words of Main Storage. It uses mainly integrated peripheral control units which are integrated into the System 11 Central Complex cabinet, and also supports several additional externally controlled disk and tape subsystems. A more detailed description of the System 11 can be found in UP-10154 Rev. 2, System 11 System Description [40].

3065-97 Small System (1 IP, 1 1,024K MSU, 1 DCC, 1 String Controller, 2 8436 Disks, 1 BBC, 1 Integrated Tape CU, 1 Streaming Tape Unit, 1 peripheral cabinet.

3065-95 Medium System (1 IP, 1 1,024K MSU, 1 DCC, 2 String Controller, Dual Access, 4 8436 Disks, 1 Printer CU, 1 0789-45 Printer, 1 BBC, 1 Integrated Tape CU, 2 U24 Tape Units, 1 peripheral cabinet.

3065-93 Large System (1 IP, 1 1,024K MSU, 2 DCC, 2 String Controller, Dual Access, 6 8436 Disks, 1 BMC, 1 Printer CU, 1 0789 Printer, 1 BBC, 1 5055 Tape CU, 2 U28 Tape Units, 1 peripheral cabinet.

3065-91 MP System (Expansion Cabinet, 2 IP, 2 1,024K MSU, 4 DCC, 4 String Controller, 8 8436 Disks, Dual Access, 1 BMC, 1 Printer CU, 1 0789 Printer, 1 BBC, 2 5055 Tape CU, 2 U28 Tape Units, Dual Access, 2 peripheral cabinet.

3065-81 Small Dyadic System (2 IP, 1 1,024K MSU, 1 DCC, 1 String Controller, 2 8436 Disks, 1 BBC, 1 Integrated Tape CU, 1 Streaming Tape Unit, 1 peripheral cabinet.

3065-79 Medium Dyadic System (2 IP, 1 1,024K MSU, 1 DCC, 2 String Controller, 4 8436 Disks, Dual Access, 1 Printer CU, 1 0789-45 Printer, 1 BBC, 1 Integrated Tape CU, 2 U24 Tape Units, 1 peripheral cabinet.

3065-77 Large Dyadic System (2 IP, 1 1,024K MSU, 2 DCC, 2 String Controller, 6 8436 Disks, Dual Access, 1 BMC, 1 Printer CU, 1 0789 Printer, 1 BBC, 1 5055 Tape CU, 2

---

[1]Variants for 50/60Hz result in a different type/feature number suffix. In general, only 60 Hz numbers are listed to minimize volume.

U28 Tape Units, 1 peripheral cabinet.

3065-75 EMC Small Dyadic System (2 IP, 1 1,024K HDMSU, 1 DCC II, 1 BBC, 1 Integrated Tape CU, 1 Streaming Tape Unit, 1 peripheral cabinet. May be expanded to a 3- or 4-instruction processor configuration. (3065-75 60hz, 3065-74 same except 50hz).

The following optional equipment may be added to the above basic system configurations. Type and feature number suffixes have been omitted due to volume.

K3653	MSU Expansion
2028	Main Cab Conversion and MSU Expansion
3066	Expansion Cabinet
K3649	Expansion Processor (IP)
F3651	2nd BBC (Byte Bus Channel)
F3714	Aux Console Control Unit
3660	Aux Console
0425	Console Printer
K3652	BMC (Block Mux Channel)
2054	EMC BMC
F3725	L Bus Power Expansion
1974	Peripheral Cabinet
K3650	DCC (Disk Controller/Channel)
K3920	String Controller
8436	Disk Drive
K3886	Disk Expansion
K3887	Dual Access
F3674	Integrated Tape Control Unit
2014	1st Streaming Tape Unit
K3782	2nd Streaming Tape Unit
0876	U22 Tape Drive
0876	U24 Tape Drive
F3851	U22 Tape Formatter
F3853	9 Track NRZI
5055	Uniservo 2X Tape Control
F2451	9 Track NRZI
F3738	Dual Channel
F3739	ASCII/EBCDIC
F3116	Dual Access
0884	U26 Tape Drive
0884	U28 Tape Drive
F3737	Dual Access
5058	U24 7-Track Subsystem
F0823	7 Track NRZI
F0825	Dual Channel
F3955	WSCU (Workstation Control Unit)
F3842	CLCU (Communications Line Control Unit)
2006	ICP 2 (Integrated Communications Processor)

F3165	Multi-Line Async Line Module
F3163	Medium Speed Loadable Line Module
F3837	Multi-Line Sync Line Module
F3847	DCSS Line Module
F3939	DCP/10 Interface
F4091	Printer Cabinet
F3672	Integrated Printer Control Unit
0719	Card Reader
0789	Printer
F3866	Remote Power Sequencing
F2865	Print Band
0776	Printer
F2346	Print Cartridge
F2347	Print Cartridge
F2217	Speed Upgrade
F2216	Print Cartridge
F2215	Print Cartridge
F4257	Disk Control Channel II (DCC II)
8451	Disk Drive
F4332	8451 Dual Access
F4329	8451 Disk Drive Expansion
8463	Disk Subsystem
F4640	Disk Drive

Additionally, A DCP/10, DCP/10A, DCP/15, DCP/20, or DCP/40 may be attached to a System 11. See Common Communications Section.

1100/90

The 1100/90 System is a large scale system which supports from 1 to 4 Instruction Processors (IPs), from 2,097K to 16,777K 36-bit words of main storage, and from 1 to 4 Input / Output Processors (IOPs). The 1100/90 I/O complex connects to numerous externally controlled peripheral devices. A more detailed description of the 1100/90 system can be found in UP-9288 Rev. 1, 1100/90 Systems, System Description [22].

3054-67 1100/91 SV Processor Complex (1 Entry Level IP, 1 Operator Console, Color CRT, Clock Calendar, Console Printer, System Clock, System Panel, 1 MSU, 1 SSP, Motor Alternator, 1 IPCU, 1 IOP, 1 BMCM, 1 WCM).

3054-99 1100/91 Processor Complex (1 IP, 1 Operator Console, Color CRT, Clock Calendar, Console Printer, System Clock, System Panel, 1 MSU, 1 SSP, Motor Alternator, 1 IPCU, 1 IOP, 1 BMCM, 1 WCM).

3150-99 1100/91 Model II Processor Complex (1 IP, 1 Operator Console, Color CRT, Clock Calendar, Console Printer, System Clock, System Panel, 1 8M Word MSU, 1 SSP, Motor Alternator, 1 IPCU, 1 IOP, 1 BMCM). 3150-97 same but with 28 inch high System Console.

The following additional equipment may be added to the above basic system configurations:

CENTRAL COMPLEX

3054-03	1100/90 Instruction Processor Expansion
2054-63	SV Instruction Processor Expansion
3150-95	1100/90 Model II Instruction Processor Expansion
F3378	IP Performance Monitor
F4088	IP External Hardware Monitoring Interface
3067	Input/Output Processor
1954	IPCU (IP Cooling Unit)
F5230	IP Single Port Feature
F5231	IOP Single Port Feature
K3675	WCM (Word Channel Module)
K3676	BMCM (Block Mux Channel Module)
F3953	FIPS I/O Compatibility
1964	IOP Expansion Cabinet
F3938	IOP Performance Monitor
7052	Main Storage Unit
7055	High Speed Main Storage Unit
K3125	MSU Storage Expansion
3562	Color Auxiliary Console
4026	Operators Console
3660	Console CRT Expansion
0429	Console Printer
1980	SPC (Subsystem Power Control)
K3728	SPC Expansion
F3729	SPC Interface Expansion
F3947	SPC / SSP Interface
8513	Motor Alternator
2414	Motor Alternator Paralleling Unit
F4168	Remote Control Panel
F4169	Remote Paralleling Panel
3058	SSP (System Support Processor)
2533	Byte Channel Transfer Switch
F4259	2533 Switch Expansion

DISK SUBSYSTEMS (Note: Dual access, 50/60 cycle, expansion features not listed due to volume).

5040	Disk Control (8430/8433/8450)
8450	Disk Drive
5056	Disk Control (8470/8480)
8470	Disk Drive
8480	Disk Drive

5057	SSD / DISK / Cache Control
7053	Disk Cache
8481	Disk Drive
5090	Disk Control (8490)
8490	Disk Drive
9494	Disk Subsystem
9399	*String Controller*

TAPE SUBSYSTEMS (Note, Dual Access, 50/60 Cycle, 7/9 Track Features, Expansion Features not listed due to volume).

5045	Uniservo 14 Control
0870	Uniservo 14
5058	Uniservo 22 / 24 Control
0876	Uniservo 22
0876	Uniservo 24
5055	Uniservo 2X Control
0884	Uniservo 26
0884	Uniservo 28
5042	Uniservo 3X Control
0872	Uniservo 30
0873	Uniservo 32
0873	Uniservo 34
0874	Uniservo 36
5061	Uniservo 36 Control
5042	Uniservo 36-II Control
0874	Uniservo 36-II
5073	Uniservo 40 Control
0899	Uniservo 40

PRINTER SUBSYSTEMS (Note: Features for printer speed differences, print bands, etc. not listed due to volume).

0770	Printer
0770	0770-II Printer
0776	Printer
0777	Online Laser Printer
9246	Printer

CARD READER SUBSYSTEM

0716 Card Reader

1100/90 UNIQUE COMMUNICATIONS

F4987 1100/90 Word Channel L/M Note: A DCP/10, DCP/10A, DCP/15, DCP/20, or DCP/40 may be attached to a 1100/90. See Common Communications Section.

2200

The 2200/200 is a midframe computer system. It can be configured with from 1 to 4 Instruction Processors, and from 2,048K to 12,288K 36-bit words of main storage. The 2200/200 IP uses CMOS III VSLI Chip Technology which minimizes the number of chips required. It includes an 8K word internal cache memory and a duplicate chip set to provide constant monitoring of the IP and to be used as the IP if a failure is detected. The 2200/200 uses mainly integrated peripheral controllers which are installed in the 2200/200 central complex. A block mux channel allows attachment of additional externally controlled peripheral devices. A more detailed description of the 2200/200 can be found in UP-11429, 2200/200 Systems, System Description [41].

3088-80 2200/201 Processor Complex (1 IP with Duplicate Chip Set, 1 MSU, 1 IOP, 1 LBA, 1 SHA, 1 Formatter II, 2 170MB Fixed Media Disk Drives, 1 WSCU, 2 SVT 1121 Terminals).

3088-96 2200/202 Processor Complex (2 IP with Duplicate Chip Set, 2 MSU, 1 IOP, 1 LBA, 1 SHA, 1 Formatter II, 2 170MB Fixed Media Disk Drives, 1 WSCU, 2 SVT 1121 Terminals).

3088-95 2200/203 Processor Complex (3 IP with Duplicate Chip Set, 2 MSU, 1 IOP, 1 LBA, 1 SHA, 1 Formatter II, 2 170MB Fixed Media Disk Drives, 1 WSCU, 2 SVT 1121 Terminals, 1 Expansion Cabinet).

3088-94 2200/204 Processor Complex (4 IP with Duplicate Chip Set, 2 MSU, 1 IOP, 1 LBA, 1 SHA, 1 Formatter II, 2 170MB Fixed Media Disk Drives, 1 WSCU, 2 SVT 1121 Terminals, 1 Expansion Cabinet).

The following additional equipment may be added to the above basic system configurations:

CENTRAL COMPLEX

F3842	Central Support Interface
3089	Expansion Cabinet
F4111	Instruction Processor Expansion
F4151	High Speed Multiply / Divide
F4073	MSU Control and Expansion
F4076	Main Storage Expansion
F4612	L-Bus Expansion Module
F4080	I/O Processor Expansion
F4257	DCC II (Disk Controller Channel II)
K3652	Block Mux Channel
F4369	BMC Conversion
F4078	L-Bus Adapter
F4077	BPA (Byte Peripheral Adapter)

F3955	WSCU (Work Station Control Unit)
F4079	SHA (SCSI Host Adapter)
F4113	Formatter II
F4115	Disk Drive
1974	Peripheral Cabinet
K3728	SPC (Subsystem Power Controller)
2014	1st Streaming Tape Drive
K3782	2nd Streaming Tape Drive
8451	Disk Subsystem
F4329	8451 Disk Drive Expansion
F4332	Dual Access
8494	Disk Subsystem
5074	8494 Disk Controller
0876	1st U22 Control and Drive
0876	1st U24 Control and Drive
0876	U22 Tape Drive
0876	U24 Tape Drive
5055	Uniservo 2X Control
0884	U26 Tape Drive
0884	U28 Tape Drive
0447	Medium Speed Laser Printer
F4223	0447 Integrated Printer Control
0789	Printer
0776	Printer
0770	770-II High Speed Printer
0716	Card Reader

2200/200 UNIQUE COMMUNICATIONS

None

A DCP/10, DCP/10A, DCP/15, DCP/20, or DCP/40 may be attached to a 2200/200. See Common Communications Section.

Common Communications

COMMON COMMUNICATIONS (Note: Not all expansion features, 50/60hz versions, etc. are listed due to volume).

1986	DCP/10
1986	DCP/10A
1986	DCP/15
2053	DCP/15
8597	DCP/20
8596	DCP/40
F1933	IOP Controller Module
F2941	Second IOP Expansion

F1932	Third/Forth IOP Expansion
F1947	Host Byte Channel L/M
F1941	Async Line Module
F1942	Sync Line Module
F3163	Medium Speed Loadable L/M
F3164	High Speed Loadable L/M
F3165	Multi-Line Async L/M
F3837	Multi-Line Sync L/M
F3847	DCSS Line Module
F4230	Twisted Pair L/M
F4235	SDM Twisted Pair Module
F3882	FEPI
8409	Disk
8441	Disk
F4158	Integrated Disk
F1939	Integrated Flexible Disk

Hardware Revision Levels

Unisys 1100 and 2200 hardware is released and controlled by system plateaus. The system plateau provides a definition of central complex hardware revision levels, central complex microcode revision levels, and diagnostic software which are tested and released together as a package.

The plateau levels for the systems to be evaluated are:

SYSTEM	LEVEL
System 11	65.5, 75.5 or 86.2
1100/90	NC 1.5 with IP 3054-03 type only
2200/200	4.01

EVALUATED SOFTWARE COMPONENTS

Evaluated Software

The OS 1100 TCB contains a central Executive with ancillary software products. The hardware base consists of the configuration supporting the Executive and the ancillary software products, as well as the SSP and the DCP. The SSP hardware supports the SSP operating system. The DCP hardware supports DCP/OS and TELCON.

The TCB is configured as follows:
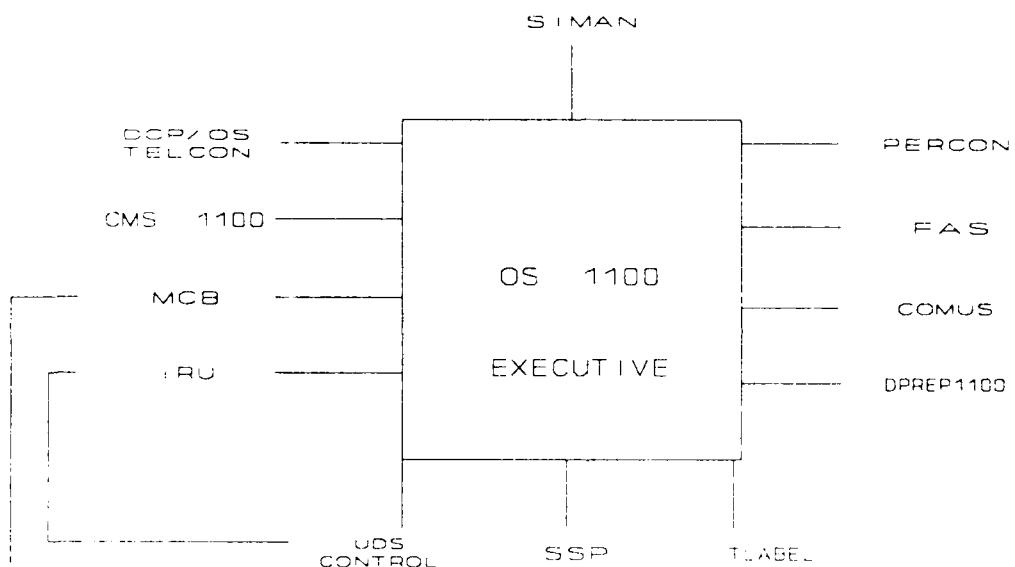
The system software which comprises the Trusted Computing Base (TCB) is the same for all hardware systems being evaluated. The initial Series 1100/2200 Operating System class B1 release is UNISYS OS 1100 Security Release I. UNISYS OS 1100 Security Release I contains the following TCB components with the indicated revision levels:

COMPONENT	LEVEL
CMS1100	5R1SE1
COMUS	5R1
DPREP	1100 4R3
EXEC	41R2S With Security Option 3 F6215-00
FAS	3R1 with PLE 10863139
IRU	5R1

LA	2R2 with PLEs 10816297, 10827213, 10832152, and 10846790
MCB	5R1S
PERCON	2R4A
SIMAN	3R1S
SSP	
SYSTEM 11	5R4
1100/90	4R7ES
2200/200	6R2
TELCON	
ICP	6R3A
DCP	7R2A
TLABEL	2R1 with PLEs 10807859, 10811422, 10876052, and 14529594
UDSC	4R1 with PLE 10816891

ACRONYMS

ACF	Access Control Function
ACR	Access Control Record
AFCB	Alternate File Common Bank
ASA	Activity Save Area
BBC	Byte Bus Channel
BDI	Bank Descriptor Index
BDT	Bank Descriptor Table
BMC	Block Mux Channel
BMCM	Block Mux Channel Module
CBSS	Common Bank Subsystem
CLCU	Communication Line Control Unit
CMS	Communications Management System
COMUS	Computerized Onsite Maintenance for User Systems
CSI	Control Statement Interpreter
DCC	Disk Controller/Channel
DCP	Distributed Communications Processor
DPREP	Disk Preparation
ECL	Executive Control Language
EMC	Electro Magnetic Control
ER	Executive Request
ESI	Externally Specified Index
FAS	File Administration System
GAP	General Access Permission
GRS	General Register Set
HDMSU	High Density MSU
ICP	Integrated Communications Processor
IOP	Input/Output Processor
IP	Instruction Processor
IPCU	Instruction Processor Cooling Unit
IRU	Integrated Recovery Utility
ISO	International Standards Organization
LA	Log Analyzer
LBA	L-BUS Adapter
MCB	Message Control Bank
MCT	Master Configuration Table
MFD	Master File Directory
MP	Multiprocessor
MSU	Main Storage Unit
PCT	Program Control Table
PERCON	Peripheral Control
PLM	Programmable Line Module
RFC	Request for Change
RSI	Remote Symbiont Interface
SAP	Special Access Permission
SCDT	Security Compartment Definition Table

27 September 1989

SHA	SCSI Host Adapter
SIMAN	Site Management Complex
SMDT	Security Mandatory Component Definition Table
SSA	System Security Administrator
SSCT	Subsystem Control Table
SSP	System Support Processor
SVT	Sperry Video Terminal
TELCON	Telecommunication Control
TIP	Transaction Processing System
TLABEL	Tape Labeling Utility
TPS	Trusted Privilege Set
UDSC	Universal Data System Control
UDS	Universal Data System
UPR	User Profile Record
WCM	Word Channel Module
WSCU	Workstation Control Unit

GLOSSARY OF TERMS

(Parentheses contain document reference and page number).

Absolute element - An executable, machine-language element produced by the Collector, the system's linker. Absolute elements are created from user-defined relocatable, machine-language, elements, together with the needed elements from the relocatable subroutine library. (16, p. Glossary 1)

Access Control Record (ACR) - A record containing a list of users, accesses they may have to an object, and the conditions under which they are allowed access. Attaching an ACR to an object makes the object semi-private.

Activity - A virtual processor. The active execution entity of the system, defined by a specific processor state, a set of register values and a program execution address. The 1100 Executive maintains the environment of an activity such that the activity appears to have continuous use of a single Instruction Processor (IP) for as long as it desires, even though the 1100 Executive may interleave processor usage among many different activities and execute them on different IPs.

Address tree - A model of the address space represented by the addressing hardware. The hardware supports a four level address tree (activity, program, application and system) although the evaluated system only supports the last three levels.

Assignment - Connection (allocation) of a file to a run.

Bank - A separate portion of contiguous virtual memory which is identified by bank descriptor registers (addressing hardware) in the Instruction Processor (IP).

Basic mode - User addressing and protection which is fully compatible with the 1100/60. The physical addressing is 16 million words; however paging instructions cannot be supported. User address space is a total of 262K words using four separate base registers. (37, p. G-2)

Bundle - To reduce the frequency of security validations and logging and to increase transaction processing performance, TIP File Security gathers files under its control into security bundles for group processing. Every TIP file belongs to exactly one security bundle. All of the TIP files in one security bundle have the same security properties. (42, p. 14-5)

C-Series - A series of hardware which includes: the 1100/90, System 11, and the 2200/200. Proper protection of common banks requires C-series hardware. (4, p. 2-2) Provides extended mode addressing and enhanced bank protection.

Cataloged file - A permanent file created by entering file information into the Master File Directory. (42, p. Glossary 2)

27 September 1989

Common bank - A portion of virtual memory that is part of the virtual address space of all users. Portions of the virtual address space are shared between subsystems to facilitate controlled sharing of information and for reasons of system efficiency. (42, p. Glossary 2)

Domain -Each lock and key value is composed of a domain number and a ring number. All banks in a subsystem have the same domain number, and this value is different from any other active subsystem that can be addressed by the original subsystem. The domain is part of the lock and key mechanism. (4, p. A-4)

DPREP - Software used to prepare and maintain disk space. DPREP1100 prepares disks for use with OS 1100. (75, p. BIB-1, 1-1) This product is used to support object reuse for disk packs.

Element - Part of a program file, usually used to store a program or subprogram. Elements can contain either symbolic images or machine-language instructions. (16, p. Glossary 7)

Exclusive assignment - Assignment of a file exclusively to a run, i.e., the file is not shared with other runs for any type of access.

Executive - The 1100 Executive is the control program for the OS 1100 for all Series 1100 and 2200 processors. As such, it is the principal software module responsible for administering the Security Policy. (4, p. 5-5) It is also referred to as the kernel.

Exec worker - an activity associated with the 1100 Executive that performs work on behalf of a user, usually as the result of an Executive Request (ER).

Executive Request (ER) - A machine instruction that allows a user to request services from the kernel.

Extended mode - An addressing mode available in C-Series hardware that provides a larger user address space and the use of 16 user base registers. An additional 16 base registers are available to the kernel. A separate field in the instruction specifies the base register to be used for address generation. Two indexing modes provide for either an 18-bit, or a 24-bit, index register value. Extended mode is required to support paging and gates. Gates support common bank protection. (37, p. G-3)

Gate - A hardware protection mechanism which is utilized by the kernel to control access to subsystems and to effect the change in execution state associated with the transfer to a subsystem. Gate processing is known as domain crossing. Access to gates as well as all code banks is controlled in a manner similar to data banks. (37, p. G-3)

General Access Permission (GAP) - An activity has general access permission to a bank when its key does not match the bank's lock, which means domain numbers are

unequal and the ring number of the activity is greater than or equal to the ring number of the bank. GAP may be read, write, enter or null. GAP is also referred to as non-owner access. (4, p. A-4)

Kernel - That portion of the TCB which operates with a processor privilege of 0.

Key - A key is associated with an active process. This key is matched against a lock before a bank can be accessed. Access keys are unique by subsystem. The 1100 Executive assigns a key value when an activity is created. (4, p. 1-5, p. A-3)

Keyin - Instructions, usually entered from a console keyboard, that tell the 1100 Executive to perform certain tasks, such as display facility status, or clear jump keys. Every
user-id has a specific console mode and set of keyin groups associated with it. The operator's console can perform all keyins. The unsolicited keyins allowed for each user-id are the intersection of the console mode and keyin groups allowcd for the user-id. (82, p. Glossary 6; 154, p. 5-30)

Lock - Governs the kind of access which is possible into a bank of memory or a gate. All banks which are owned by a subsystem receive the lock associated with that subsystem. The 1100 Executive issues the lock when the subsystem is installed. Gate locks reside in the gate while memory locks reside in memory bank descriptors. (4, p. 1-5, p. A-3)

Log - The system log file is the mechanism used to support the audit trail where security relevant events are recorded. A configuration parameter exists which forces the system to halt if the log becomes unavailable. (42, p. 15-1)

Non-owner access - Files can be accessed by subjects which are not their owners (creators) in two ways. One, a file can be declared public by its creator. Two, a file can be semi-private with an ACR attached. (42, p. 9-2)

With reference to memory access, Unisys informally uses this term as an equivalent to General Access Permission (GAP).

Owner access -A file owner is the user-id associated with the creator of the file. The owner of a file obtains all access privileges. (4, p. 3-3)

With reference to memory access, Unisys informally uses this term as an equivalent to Special Access Permission (SAP).

Private - An access state for an object in which access is only allowed to the owner.

Privilege - A means for a trusted user to override the inherent controls provided by the kernel. Privileges are given to specific users by the SSA. A user's privileges are maintained in the user-id security record.

Processor Privilege - A value maintained as part of activity state that specifies the activity's ability to execute privileged instructions and access the exec register set.

Program file - A type of mass storage file formatted for storing programs. The programs within program files are contained in elements. To keep track of the elements, each program file has a table of contents. (16, p. Glossary 16) Access is controlled at the file level.

Project-id - An identifier associated with each run. It may be used as a grouping mechanism for ACRs. (42, p. Glossary 6)

Protected objects - Protected objects are files, tape volumes, and subsystems. (4, p. 3-5)

Public - An owned object type which may be accessed by any user where the security attributes of the user equal the security attributes of the file. The owner has all access while all other subjects have read, write, delete, and execute access. (42, p. 9-4)

Qualifier - A means of ensuring that file names are unique even though they may have received the same name from the creator of the file. User-ids and project-ids are commonly used as qualifiers, but others can be created. The qualifier can be up to 12 characters. (16, p. 3-14)

Ring - A part of a lock and key value which determines how privileged an activity is, in relation to memory access, or how sensitive is the information of a bank. It is this part of the lock and key mechanism which determines whether an activity may access a bank when they are in different subsystems. (4, p. A-4)

Run - A sequence of tasks that users link together to form a self- contained unit of work. Users control a run by using a series of ECL statements that tell the operating system what to do. A run begins with the RUN statement and ends with the FIN statement. (42, p. Glossary 6)

Secured ER - Those Executive Requests (ER) which need to be controlled to maintain security or system integrity. A secured ER can only be executed by a user whose security record specifies the appropriate execute privilege. (42, p. B-1)

Security database - The set of system files, created by the 1100 Executive, which hold information about users and objects, their identities, security attributes, ownership, access identification (ACRs) and user resource maximums. (42, p. 6-1)

Security-privileged system high - A subject whose security attributes include a clearance level range of 0-63, the system compartment set (ALL), and the privileges SSBYCOMP, SSCCL, and SSBYPASSOWNR. (42, p. Glossary 7)

Semiprivate - An access state for an object which indicates that access to the object is controlled with an Access Control Record (ACR). (42, p. 9-3)

Special Access Permission (SAP) - An activity has special access permission to a bank when its key matches the bank's lock which means that either the domain numbers are equal or the ring number of the activity is less than the ring number of the bank. (4, p. A-4) Also referred to as owner access.

Subadministrators - Personnel who only manage user security records. (42, p. 7-1)

Subsystem - A subsystem defines a particular protection environment along with other attributes which are not applicable to security. (4, p. 3-6) The attributes of a subsystem are taken from the owner of the file containing that subsystem.

System Compartment Set - All compartments defined in the system. Implemented by the ALL flag in the subject, or object, security record. For an object, any new compartments defined for the system are associated with this object. For a subject, the subject will have access to any new compartment defined in the system.

System high - When referring to a subject, a subject which executes with a clearance level of 63 and the system compartment set (ALL). When referring to an object, an object that has a clearance level of 63 and the system compartment set (ALL) specified in its security record. (42, p. Glossary 8-9)

System low - When referring to a subject, a subject which executes with a clearance level of 0 and the NULL compartment set. When referring to an object, an object that has a clearance level of 0 and the NULL compartment set specified in its security record. (42, p. Glossary 9)

Tape volume - A tape volume is a single reel of tape. (4, p. 3-6)

Task - The execution of a program or transaction. A task is a discrete processing step in a batch or demand run or a TIP session involving the execution of an absolute element. (4, p. A-1)

TIP File - A partition of a file, wholly contained in that file, known to, and processed by, the Transaction Processing System (TIP). Access is controlled at the level of the containing file.

Trusted - Referring to a TCB component or a subject that has the ability to violate the security policy but does not. (42, p. Glossary 9)

Trusted Privilege Set - If a subsystem has any one of this specific subset of the security relevant privileges, then that subsystem is defined as a trusted subsystem. The security policy defines different controls for a trusted subsystem than for a non-trusted subsystem.

Trusted subsystem - Any subsystem granted privilege to violate the protection philosophy. Whenever a trusted subsystem is created it is created to run at the maximum level of its associated user-id. (4, p. 1-4)

Unclassified - A object with a clearance level of 0 and the NULL compartment set. (42, p. Glossary 10) Also referred to as system low.

User - A user is a person who requests access to the system. (4, p. 3-5)

TEAM TESTS

This section contains descriptions of the team tests performed by the formal evaluation team:

01. COMUS Installs without Privilege

>   Purpose: Verify that COMUS does not allow an unprivileged user do an install.

>   Procedure: Use an unprivileged or partially privileged user-id and attempt to install a common-bank subsystem (PERCON, Telcon, COMUS, IRU, UDSC, MCB, SIMAN, CMS1100).

>   Results: An unprivileged user attempted to install a common-bank subsystem in an existing user-id and aborted.

02. Inspection of Common ER Code

>   Purpose: Verify that certain ERs share common security-relevant code and that the "gray-box" approach, used by Unisys, is an appropirate method for reducing test cases.

>   Procedure: The team visually inspected the code for ERs which were typically related to I/O to determine whether or not specific ERs shared common security relevant code.

>   Results: ERs: PRINT$, APRINT$, PRNTA$, and APRNTA$ all share common security-relevant code. ERs: READ$, AREAD$, READA$, and AREAD$ all share security-relevant code. ERs: PRTCN$, APRTCN$, PRTCA$, APRTCA$, PCHCN$, APCHCN$, PCHCA$, APCHCA$, PUNCH$, APUNCH$, PNCHA$, APNCHA$, SYMB$, TREAD$, and ATREAD$ all share common security-relevant code.

03. Illegal ERs

>   Purpose: Demonstrate that the Executive only recognizes ERs specified as legal ERs. Both ERs documented as disabled for B1 as well as ER octal codes which are not documented as functioning will be tried.

>   Procedure: Establish a system-high user as the user identified in the following run. A MASM program will be written whereby a process is spawned which declared a table containing the following literal:

>   ER E   72 zero-fill

>   This test will be repeated 1-7777 (octal) times.

Expected Results:

A program was written that attempted each possible ER. The program correctly aborted in all cases expected: ERs requiring privilege, undefined ERs within the legal range, and ERs outside of the legal range.

04. System Operator and CMS Operator Keyins

Purpose: Verify that all operator keyins are audited in an adequate manner.

Procedure: After a week of testing, the team reviewed the continuous console log and identified specific console commands. Then the team performed an anlysis of the log to inspect that the correct entries had been made on the log. To validate logging CMS console commands, all CMS operator commands were attempted at the console. Subsequently, the log was inspected to verify all CMS entries had been logged correctly.

Results: The team examined log reports using the Log Analyzer tool and determined that all console entries had been logged correctly.

05. Illegal and Privileged Op Codes (also Paging Instructions on 2200/200)

Purpose: Determine the system state when an unprivileged user issues illegal or privileged hardware instructions (op codes)

Procedure: Unisys provided this test for this team, since they believed the team felt that this information had not been previously discussed and Unisys had not previously performed such test. The team reviewed the Unisys code and results for this test. Unisys created a table which included each op code combination (16383 possible combinations) and the expected results for each of the three system types. Any one of the following five types of results were possible:

- Instruction was legal and executed normally
- Instruction was legal but generated a reference violation
- Instruction was legal but generated an ER interrupt to the EXEC
  which was out of range (illegal subfield values)
- Illegal instruction traps to illegal operation
- Instruction was privileged and trapped to illegal operation.

This test was run on each system type under evaluation by the team.

Results: All attempted machine operations performed as expected.

06. Administrator/Subadministrator SIMAN Tests

Purpose:

1) Verify that batch and demand interfaces into SIMAN can be used to create administrator, subadministrator, and user accounts.

2) Spot check SIMAN Manual and TFM.

Procedure:

1a) Interactively create an administrator from the Master Account.

1b) Create a subadministrator with a subset of privileges from the administrator account created in step 1a.

1c) Create user accounts with subsets of privileges from the subadministrator account created in step 1b.

1d) From each created account, use SIMAN to verify that privileges specified in account creation actually were the ones conferred in the new account.

2) Repeat steps 1a through 1d using the batch interface.

Results:

1) SIMAN manual and TFM instructions were sufficient to perform the the test.

2) Tests did not cause error conditions in either user interface. The batch interface imposes the exact same restrictions as the full-screen interface.

3) Privileges specified at user-id creation are manually verified from the created account.

4) SIMAN only allows access to information on user-ids and accounts as noted in the SIMAN manual and the TFM (see below)

The procedures outlined in the test were followed, and the below user-id tree was created:

```
               SSA
              /   \
          SUBA    SUBB
          /          \
     SUBSUBA      USERA USERB
       /
    USERC
```
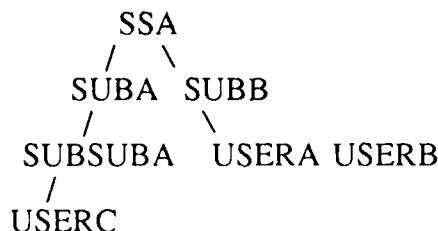
When this tree was created, SUBA could not view USERC statistics. Each subadministrator could only view the user-ids that it created. When SUBSUBA was deleted, SUBA inherited administrative control of USERC.

07. Writing to PCT (modifying address space and status word)

Purpose: Determine if a user can modify the contents of his PCT.

Procedure: Unisys modified their "Reference Violations and addressing Exception Interrupts" test (F58) to include an attempted write to a user's PCT in the manner the team was going to. The test code was examined to assure that a write was attempted and the test results were examined and mapped back to the test to assure the results were a result of the attempted write.

Results: Process did abort.

08. St. Paul Debug Code

Purpose: Ensure that the SSP, in system mode, cannot access memory.

Procedure: The team attempted to access memory through the SSP while it was in system mode by using SSP console keyins.

- /TRACE SPDEBUG allows access to STPAUL debug code, in the SSP, if the code exists. The team verified this on a non-B1 system.
- With STPAUL debug code in the SSP, Partitioning Maintenance Control Bits must be set which place a specified module of main memory in Maintenance Mode and grants access to that module from the SSP while in System Mode. The SSP Partitioning Maintenance Control Bits cannot be set if the SSP is in System mode and the STPAUL debug code is absent. The team performed all the above tests.

Results: The team verified that no STPAUL debug code existed in the B1 system.

09. Console Privilege Inheritance

Purpose: Verify the description on console capability (&&cons) as described in the B1 documentation.

Procedures: In the last update to Unisys provided test, a test was provided which created six user-ids with the following associated console privileges:

USER-ID /	CLR-	COMPARTMENT SET		CONSOLE	
PASSWORD	LEVEL	Maximum	Default	PRIVILEGES	MODE
CK-NON/PASS01	0-63	all	all	all	None
CK-BAS/PASS02	0-63	all	all	all	Basic
CK-LIM/PASS03	0-63	all	all	all	Limited
CK-FUL/PASS04	0-63	all	all	all	Full
CK-DIS/PASS05	0-63	all	all	all	Display
CK-RES/PASS06	0-63	all	all	all	Response

Every console keyin was attempted from a terminal assigned a particular console privilege which must be specified by SIMAN (default = none).

Results: A demand terminal is only allowed to perform commands appropriate for the level of console privilege conferred.

10. Verify Software Configuration on the test systems.

Purpose: Verify that the proper software versions are installed on the test systems.

Procedure: The team "builds" a B1 system and uses the resulting boot tape to bring the test systems up and finish installing the system products. After the B1 system was up, a runstream was executed to produce a list of the product versions actually installed on each of the test systems used by the team. These lists were then compared with the list of products which the vendor provided the team (see Appendix B).

Results: The list of installed product versions on the test systems agreed with the list provided by UNISYS.

11. Three specific Exective Request - EX$CRD, SMU$, USER$

Purpose: Verify the proper functioning of these three Executive Requests in a B1 environment. EX$CRD and SMU$ contain code which should not be active on a B1 system, while USER$ contains code which should function only for the Site Administration Tool (SIMAN).

Procedure: The team executed a MASM program which submits ER packets to each ER. In all cases, the ER should indicate that the function was not performed. EX$CRD should indicate that the ER is not in use. USER$ and SMU$ should return "000400000000" in a snap of the registers, indicating the function was not performed.

Results: The MASM programs were assembled and run. They produced the expected results. The programs were run under the Security Officer's account to show that a this account does not confer undocumented privilege.

12. Verify Non-configuration of local printers.

Purpose: Ensure local printers are not configured by default when the B1 system is brought up.

Procedures: Unisys configured a local printer to a terminal to demonstrate the whether or not the terminal output could be directed to the printer. The team could not assign the device.

Results: Test demonstrated that local printers can not be attached.

13. Dynamic reconfiguration of consoles.

Purpose: Ensure that only preconfigured devices may be designated as consoles.

Procedure: The team learned that all consoles have to be identified to the system through SIMAN at system generation time. The only dynamic reconfiguration allowed is within this identified set of consoles.

Results: Unisys successfully demonstrated that reconfiguration is only possible within a specified set of consoles identified at system generation time.

14. Change DAC on tape.

Purpose: Domonstrate the feasibility of a user changing DAC on a tape volume.

Procedure: ACRs are not written to tape. Rather the name of an attached ACR is written in the VOL3 tape header. The actual ACR is in the security file SYS$*SACRD$. The owner of the ACR is granted access exactly as an owner of an ACR pointing to a semi-private disk file.

Results: Unisys provided sufficient information to verify the proper controls over ACRs associated with tape files.

15. Write VOL3 header on a tape prelabeled with VOL1.

Purpose: To verify that prelabelled tapes with the serial number, owner and security attributes in the magnetic label is controlled by the system in the same manner as prelabelled tapes with only the serial number and the owner identified in the magnetic label. The magnetic label begins with a VOL1 record which includes the name of the owner and the serial number of the tape volume. The next record in the header is a VOL3 record which includes all security information (ACR name, classification, compartment set etc.). Unisys does not use a VOL2 record which is optional in FIPS required labelling. ER TLBL$ only creates VOL1. ER TVSLBL$ creates a VOL1 and VOL3 record.

Procedure: The team verified that when a user attempted to write to a tape volume,

the correct MAC and DAC were in the label regardless of which ER was used to prelabel the tape. The team also attempted to backspace the tape into the label area (first file) to determine whether or not a user could modify label contents.

Results: If a tape is prelabelled with ER TVSLBL$, the MAC and DAC of the user must match or dominate the contents of the label for access. If the tape was prelabelled with ER TLBL$, then, if the user-id on the tape (VOL1) matches the user of the requesting subsystem, a VOL3 record with security attributes will be written on the first tape write. The security attributes are taken from the requesting subsystem. It is possible to backspace into the physical label area of a tape (reverse read and reverse backspace one file). However, whenever an IO operation was attempted, the tape was rewound to load point, the label read and the tape positioned after the label.

This page intentionally left blank.

REFERENCES

[1] Dockmaster interactive meeting "Sperryendor", transaction number 718, PROPRIETARY.

[2] Security Ownership Addendum, CRFC 1853, 11/30/83, Sperry Univac Computer Systems, PROPRIETARY (Inventory: 4)

[3] B1 Security-- DoD Developmental, briefing slides, March 5, 1986, Sperry Corporation, PROPRIETARY. (Inventory: 25)

[4] Unisys 1100/2200 TCB Security Design, ESD-201, July 21, 1989, Unisys Corporation, PROPRIETARY. (Inventory: 198)

[5] Automatic Enforcement of New Secured Privileges and ERs, SFD 2214, October 22, 1986, Unisys Sperry Computer Systems, PROPRIETARY. (Inventory: 31)

[6] Mandatory Security Attributes of B1 System Files, SFD 2210, Oct 13, 1986, Unisys Sperry Computer Systems, PROPRIETARY. (Inventory: 32)

[7] Unisys OS 1100 Compliance with B1 Criteria, April 29, 1987, Unisys Corporation, PROPRIETARY. (Inventory: 35)

[8] Unisys OS 1100 B1 Security Program Overview, April 29, 1987, Unisys Corporation, PROPRIETARY. (Inventory: 37)

[9] Product Response Statement (PRS#57) Trusted Systems, 1/26/87, Unisys Corporation, PROPRIETARY. (Inventory: 38)

[10] Security Attribute Specification for F-cycles, C/TRFC 2136, 6/2/8 Sperry Computer Systems, PROPRIETARY. (Inventory: 39)

[11] Schedule Queue Trailer Extension, C/TRFC 2119, 3/10/86, Sperry Computer Systems, PROPRIETARY. (Inventory: 41)

[12] Disabling COMPOOL Msg. I/F, C/TRFC 2179, 9/8/86, Sperry Computer Systems, PROPRIETARY. (Inventory: 42)

[13] C2 Security Adapts- Communications Systems, CRFC 24, 1/14/87, Unisys Corporation, PROPRIETARY (Inventory: 43)

[14] TIP Session Control Recovery, RFC IRU10, 1/19/87, Unisys Corporation, PROPRIETARY. (Inventory: 45)

[15] PFS-81, Update 1, COMUS "Security Gate Bank Installation," 5/20/87, Sperry Corporation, PROPRIETARY. (Inventory: 46)

[16] OS 1100 Exec System Software, Executive Control Language (ECL) Operations and Programming Reference Manual, UP11563.5, November 1988, Sperry Corporation. (Inventory: 185)

[17] UP-9954.5, OS 1100, Site Management Complex, SIMAN Administration and Operations Guide, Level 3R1, (no printed date) (sent 11/30/87). (Inventory: 150)

[18] UP-13011, OS 1100, Security System, Functional Overview, October 1987. (Inventory: 135)

[19] "Sperry Univac Telcon System: System Description," UP-8455.Rev 2, Sperry Corporation, 1981. (Inventory: 58)

[20] "Sperry Univac Series 1100 Operating System EXEC Level 38R1: Installation Reference," UP-8486.5, Sperry Rand Corporation, 1982. (Inventory: 59)

[21] UP-8486.14, OS 1100, Exec System Software, Administration, Programming, and Support Reference Manual, October 1987. (Inventory: 142)

[22] "Sperry Univac 1100/90 Systems Hardware: System Description," UP-9288, Sperry Corporation, 1982. (Inventory: 60)

[23] "Series 1100 Transaction Processing EXEC Level 39R2: Programmer Reference," UP-8296.8, Sperry Rand Corporation, 1984. (Inventory: 62)

[24] UP-10098.2, OS 1100, Exec System Software, Transaction Processing, Programming Guide, October 1987. (Inventory: 140)

[25] "Series 1100 Systems USE-SUAE Conference Notes," Sperry System Products," Sperry Corporation, 1985. (Inventory: 64)

[26] "Fall Conference: Technical Papers," Use Inc., Houston, TX, Nov, 1982. (Inventory: 65)

[27] "Fall Conference: Technical Papers," Volume 1, Use Inc., Orlando, FL, Oct, 1983. (Inventory: 66)

[28] "Fall Conference: Technical Papers," Volume 2, Use Inc., Orlando, FL, Oct, 1983. (Inventory: 67)

[29] "Conference Proceedings," Volume I, Use Inc., Anaheim, CA, Nov, 1985. (Inventory: 68)

[30] "Conference Proceedings," Volume II, Use Inc., Anaheim, CA, Nov, 1985. (Inventory: 69)

[31] "Conference Proceedings," Volume III, Use Inc., Anaheim, CA, Nov, 1985. (Inventory: 70)

[32] OS 1100 Exec Test Plans, June 1988, Unisys Corporation, PROPRIETARY. (Inventory: 173)

[33] B1 Security Test Plan (Revision 11), August 29, 1989, PROPRIETARY. (Inventory: 202)

[34] 1100/90 Processor and Storage, UP-9667. (Inventory: 82)

[35] Type 3065 Processor and Storage, UP-9955. (Inventory: 83)

[36] 2200/200 Processor and Storage, UP-11275. (Inventory: 84)

[37] ASA-0113, C-Series Instruction Processor Architectural Specification. (Inventory: 85)

[38] ASA-0114, C-Series Chipset Instruction Processor Architectural Specification. (Inventory: 86)

[39] 1100 Systems Concepts, Student Guide. (Inventory: 87)

[40] System 11 System Description, UP-10154. (Inventory: 88)

[41] 2200/200 System Description, UP-11429. (Inventory: 89)

[42] Security System: Planning and Administration Reference Manual, UP-11572.5, August 1989. (Inventory: 200)

[43] SIMAN UAM-5, Password Enhancements PROPRIETARY. (Inventory: 91)

[44] SIMAN UAM-7, Profile Enhancements PROPRIETARY. (Inventory: 92)

[45] SIMAN UAM-8, Batch Environment PROPRIETARY. (Inventory: 93)

[46] SIMAN UAM-11, TSS Version Number PROPRIETARY. (Inventory: 94)

[47] SIMAN UAM-13, Migration Utility PROPRIETARY. (Inventory: 95)

[48] SIMAN UAM-16, Private/Public Userids PROPRIETARY. (Inventory: 96)

[49] COMUS 107, COMUS Security PROPRIETARY. (Inventory: 97)

[50] SIMAN UAM-014, Symbolic Clearance Levels, 7/31/87 PROPRIETARY. (Inventory: 98)

[51] RFC 29, Common Bank Security, Comm Systems, 8/22/86, Updated May 8, 1987

PROPRIETARY. (Inventory: 99)

[52] TLABEL-001, TLABEL 2R1 Tape Volume Security, 4/21/87   PROPRIETARY. (Inventory: 100)

[53] RFC 2036, Security Compartments - Management & Control, Rev-DRBA, June 4, 1987 PROPRIETARY. (Inventory: 101)

[54] RFC 2140, PERCON - Print Labeling for B1 Security, 7/21/87 PROPRIETARY. (Inventory: 102)

[55] SFD 2236, Disable of @FILE, 8/4/87 Rev B PROPRIETARY. (Inventory: 103)

[56] SFD 2220, Exec Symbionts - Support for B1 Security, 7/27/87 PROPRIETARY. (Inventory: 104)

[57] SFD 2231, TIP File Security, 7/23/87 PROPRIETARY. (Inventory: 105)

[58] SFD 2232, TIP Message Security, 7/10/87 PROPRIETARY. (Inventory: 106)

[59] SFD 2241-Rev P6, Security Object Attribute Compare Function, 7/29/87 PROPRIETARY. (Inventory: 107)

[60] SFD 2247, MSCON$ Privilege for DBACK$ & DBACK$1, 3/24/87 PROPRIETARY. (Inventory: 108)

[61] SFD 2251, Disabling Password Solicitation for RS & ST Keyins 4/24,87 P2 Update PROPRIETARY. (Inventory: 109)

[62] SFD 2257, Security Logging in a B1 System, 7/6/87 PROPRIETARY. (Inventory: 110)

[63] SFD 2260, Label Integrity for TIP Print$ Files, 5/87 PROPRIETARY. (Inventory: 111)

[64] SFD 2271, Security Validation for Imported Objects (R1), 6/19/87 PROPRIETARY. (Inventory: 112)

[65] SFD 2283, QI$CON Message Retrieval Privilege, 6/19/87 P1 Update PROPRIETARY. (Inventory: 113)

[66] UP-10063.3, OS 1100 Exec System Software, Common Banks Operations and Programming Guide, Technical Review, July 15, 1987. (Inventory: 114)

[67] LA-1, Security Userid Report PROPRIETARY. (Inventory: 115)

[68] LA-2, Security Object Report PROPRIETARY. (Inventory: 116)

[69] FAS, B1 Security PROPRIETARY. (Inventory: 117)

[70] UDS Overview Information, 8/10/87 PROPRIETARY. (Inventory: 119)

[71] RFC 2035, Establishing the User Profile, Rev G PROPRIETARY. (Inventory: 120)

[72] RFC 2133, Security Labeling Kernel, Rev B, Dec 29, 1987 PROPRIETARY. (Inventory: 121)

[73] RFC 2148, Inhibit Terminal Output, April 87 PROPRIETARY. (Inventory: 122)

[74] RFC 1991, Sec Enhancements for Sign-on Validation, Oct 85 PROPRIETARY. (Inventory: 123)

[75] UP-11624.2, OS 1100, Disk Preparation (DPREP1100), Operations and Support Guide, October 1987. (Inventory: 149)

[76] RFC 2020, TIP Session Control, Jan 15, 1986 PROPRIETARY. (Inventory: 125)

[77] RFC 2087, Tape Volume Security, Mar 87 Update, Aug 7, 1987 PROPRIETARY. (Inventory: 126)

[78] UP-9953.4, Series 1100 Guide to Universal Data System Control (UDS Control), Level 3R1. (Inventory: 128)

[79] UP-9347.3, Series 1100 Log Analyzer, LA Level 2R1, User Reference. (Inventory: 129)

[80] UP-10700.18, OS 1100, Exec Level 41R2, Release Description, October 1987. (Inventory: 134)

[81] OS 1100 Exec System Software, Executive Requests Programming Reference Manual, UP-4144.38, November 1988. (Inventory: 186)

[82] UP-7928.14, OS 1100, Exec System Software, Operations Reference Manual Level 41R2, October 1987. (Inventory: 138)

[83] UP-10722.5, OS 1100, Transaction Processing, Exec Level 41R2, Installation Guide. (Inventory: 139)

[84] UP-10097.3, OS 1100, Exec System Software, Transaction Processing, Administration, Operations, and Programming Reference Manual, October 1987. (Inventory: 141)

[85] UP-8448.16A, OS 1100, Exec System Software, Installation and Configuration Guide, September 1987. (Inventory: 143)

[86] UP-9953.5, OS 1100, Universal Data System Control (UDSC), Technical Overview, November 1987. (Inventory: 146)

[87] UP-10099.4, OS 1100, Universal Data System (UDS), B1 Security, Installation Guide (no date). (Inventory: 147)

[88] UP-12030.2, OS 1100, Universal Data System (UDS), Administration. Operations, and Support Reference Manual, November 1987. (Inventory: 148)

[89] UP-10069.1, OS 1100, Exec System Software, System Log, Operations, Programming, and Support Reference Manual, November 1987. (Inventory: 151)

[90] ESD-1.1, Unisys 1100 Executive Systems, ESD Process Guide, October 9, 1987, PROPRIETARY. (Inventory: 152)

[91] OS 1100 Security System Operations Guide, UP-13011.1, August 1989. (Inventory: 201)

[92] Log Analyzer LA Level 2R2, Administration and Operations Reference Manual, UP-9347.4, November 12, 1987. (Inventory: 145)

[93] Department of Defense Trusted Computer System Evaluation Criteria, Department of Defense Standard, DOD 5200.28-STD, December 1985.

[94] Dockmaster interactive meeting "Sperryendor", transaction number 736.

[95] Dockmaster interactive meeting "Sperryendor", transaction number 861.

[96] DCP Series Distributed Communications Processor Operating System (DCP/OS): Operations Reference Manual, UP-11541-A, November 1987.

[97] Telcon Level 7R1 Operator Reference, UP-9256.7, April 1987.

TEST DESCRIPTIONS

This appendix identifies how the test suites, provided by Unisys, adequately describe sufficient tests to demonstrate that OS1100 provides the required security features for a B1 system. "Adequately tested" means that test were performed for every user interface into the TCB. The words, "provides the required security features for B1", means that for each particular user interface, the security feature(s) supported by that interface were tested.

Unisys chose to label an individual test suite with the letter "F" followed by a number between one and sixty-seven and, for some test suites, this was followed by a letter subscript. Each suite of tests is called a test case, by Unisys, but not all of them describe an individual test. For instance test 8A performs a suite of tests for each of four users. One user has a system-high clearance level while the other three have a range of sixty-two clearance levels. All four users have three account numbers and four project-ids. File access to seven different files is evaluated for all combinations of user-id/clearance level/account number/project-id. The types of access tested are the ECL statements: XQT, ASG, and CAT and the ERs: IO$, IOW$, and SUMOD$. The expected results of "test" 8A required six person hours to evaluate while "test" 12 took 10 person hours to evaluate. Lower numbered tests were attempts to test an entire B1 requirement and these test typically are complex and difficult to verify. Higher numbered tests typically test one user interface into the TCB and are much easier to evaluate.

The following sections are presented in a particular order. Each section contains table(s) of information necessary to describe the information pertinent for the section. Section G-1 identifies the types of user interfaces into the TCB. This section also includes the conditions under which that interface type is tested as well as the test numbers which exercise the specific interface type. Section G-2 identifies each specific user interface into the TCB. This section also provides a reference for the documentation of the interface as well as the number of the test which completely or partially test the interface. The next section (G-3) provides a table of the specific hardware configurations for each of the evaluated models. Section G-4 identifies the suite of tests provided by Unisys by test suite number. This section also identifies the TCB interfaces explicitly tested which are providing security features as well as those interfaces which are implicitly tested simply because they need to work correctly for the test to complete successfully, even though for the specific test they are not providing security relevant decisions. Section G-5, identifies test F1-F7 which are the steps required to build and install an OS 1100 B1 product. Section G-6 identifies all TCSEC requirements for the B-1 class of systems and this table maps each requirement to the actual test number. The last section, G-7, is a table of the test results when the evaluation team ran the Unisys provided suite of tests. A summary of the sections documenting the testing perfomed by the evaluation team follows:

- Section G.1 TCB Interface Types
- Section G.2 TCB Interfaces Identified
- Section G.3 Specific Hardware Tested
- Section G.4 Test Coverage

- Section G.5 Steps to Build and Install the B1 Product
- Section G.6 TCSEC Requirements Coverage - Section G.7 Test Results

G.1 TCB Interface Types

This section consists of a table which identifies the types of user interfaces into the TCB as well as the test number which exercises the particular interface type.

TCB INTERFACE TYPES

TCB Interface	Conditions Tested	Test #
Console keyins	Audit	F6, F21
$$commands	Sign-on required	F38
Sign-on	I & A, Audit	F8A, F9, F15C, F31, F36
ECL commands	I & A, MAC & DAC, Object Reuse Labeling, Audit, System Architecture	F8A & B, F9, F11. F11B, F12, F16, F17, F21, F28, F30, F31, F32, F33, F35, F36, F37, F38, F59
@<processor>	MAC & DAC	F8A
@XQT	MAC & DAC, Object Reuse	F8A, F11A, F11D
TIP Transaction	MAC & DAC, Object Reuse	F11E, F15C
@@commands	Sign-on, System Architecture	F16, F21, F36, F38
ER calls	MAC & DAC, Object Reuse, Labeling, Audit, System Architecture	F8A, B & C, F9, F10, F11A, F11B, F11D, F12, F15C, F15D, F18, F20, F26, F28, F39,F40, F41, F42, F43, F44, F45, F46,

		F47, F48, F49, F50, F51, 52, F53, F54, F55, F61
LBJ	MAC & DAC	F10, F15A, F25, F37, F58, F67
Exec Interrupts	System Architecture	F57, F58
MCB $Action commands	Controlled usage	F64
User Commands to COMUS	System Architecture	F2, F10
User commands to FAS	MAC & DAC, file labels	F4, F14
User commands to IRU	MAC & DAC, system architecture	F6, F60
User commands to SIMAN	system architecture, I & A, LBJ	F56, F22, F23, F24, F27
User commands to TLABEL	file labels	F6, F8, F14
User commands to UDS	MAC, system architecture, LBJ	F25

G.2 TCB Interfaces Identified

The TCB interfaces include: ECL statements (job control language), @@commands (immediate interactive commands), ERs (Executive Requests), and other.

Note that the reference to the Test # lists tests that explicitly test the security controls. Also listed in the Test # column are those tests that use the TCB interface, and whose success depends on the correct functioning of the interface.

G.2.1. Executive Control Language (ECLs)

The security-relevant ECL commands are listed below along with the B1 test numbers that validates the security controls for the ECL command. Those ECL commands not listed are considered not to be security relevant.

The ECL commands are documented in the ECL Reference Manual, UP- 11563.5

EXECUTIVE CONTROL LANGUAGE

ECL Command	Description	Test #	Security Control
RUN	batch/demand job statement	F31, F56	user-id, project-id, account number, and generation control
XQT	put program into execution	F8A, F11A, F11D, F30	File access; residue; NPE disabled
processor execution	put program into	F8A	File access
ADD	alternate job stream	F59	file access
BRKPT	alternate print file	F59	file access
PMD	print program dump	F35	only user's memory visible
START	start separate job	F10, F34	file access
SYM	queue print file	F59 , F16, F12	file access; @SYM to terminals and userids is disabled on B1; @SYM Every-Page-Label Options
PASSWD	password for batch; chang for demand; comp. ment set for batch	F36	User-id validation
LEV	change Clearance Level	F33	Privilege required
ASG	assign new or existing file	F8A, F8B, F9, F10, F11B, F13A F13B, F32 F66	file access checks; privilege required for arbitrary device and 'J' option tape assign

CAT	create new file	F8A, F9, F10, F66	File security attributes
FREE	delete file	F8A, F9	Delete file access check
FILE	create data file	F17	disabled in B1

ENDF removed because it is not security relevant

CKPT	Checkpoint job	F37	disabled in B1
CKPAR	partial checkpoint	F37	disabled in B1
RSTRT	checkpoint restart	F37	disabled in B1
RIP	CKRS processor call	F37	disabled in B1

G.2.2. Transparent Control Statements (@@ Commands)

The security-relevant @@ commands are listed below along with the B1 Test number that validates the security controls for the @@ command. Those @@ commands not listed are considered not to be security relevant.

The @@ commands are documented in the ECL Reference Manual, UP-11563.5

TRANSPARENT CONTROL STATEMENTS

@@ Command	Description	Test _#_	Control Mechanism
@@ASG	Assign a new or existing file assign a device		Same as @ASG
@@BRKPT	Divert symbiont output; close or partition symbiont files		Same as @BRKPT
@@CAT	Create a new file		Same as @CAT & @ASG.CP
@@CKPAR	Checkpoint a task	F37	Disabled on B1
@@CKPT	Checkpoint a run or task	F37	Disabled on B1

@@CONS	Remote console interface; process an operator keyin	F21	User profile
@@FREE	Delete a file		Same as @FREE, delete file security attributes
@@HDG	removed because it is not security relevant		
@@LEV	Change executing clearance level		Same as @LEV
@@PASS @@PASSWD	Change password	F36	Current valid password must be specified
@@SEND	Display a print file that is queued to user-id or site-id	F16	Sym to user-id or site-id not allowed on B1
@@START	Start a separate run		Same as @START
@@SYM	Queue a print file	F16	@@SYM to terminals and user-ids is disabled
@@TERM	Terminate the current run and terminal session	F38	re-validation required
@@TM	Send text to another user or terminal	F16	Disabled on B1

G.2.3. Executive Requests (ERs)

The security-relevant ERs are listed below along with the B1 Test numberthat validates the security controls for the ER. Only those ERs that are considered to be security relevant are listed.

Some ERs are used exclusively by the TCB products to communicate with the Exec, and as such are not available to any other user. Direct testing of these intra-TCB ERs is limited to demonstrating that they are indeed controlled with privilege. In addition, some ERs are controlled with privilege, but are not used in B1 systems; testing demonstrates this control by privilege.

EXECUTIVE REQUESTS

ER	Description	UP Ref	Test #	Security Control
IO$ 1	I/O immediate	4144	see IOW$ [5]	read/write access checked
IOI$ 2	immediate I/O with interrupt	4144	see IOW$; F11A	read/write access checked; GRS initialized
IOW$ 3	wait for I/O	4144	F8A F8B, F9, F10, F63	read/write access checked; no tape read past EOF
COM$ 010	Communicate with operator	4144	F41	routing functions privileged; some functions Exec only
FORK$ 013	Create a new activity	4144	F11A	GRS initialized
TFORK$ 014	Create a new activity; timed wait	4144	F11A	GRS initialized
PRINT$ 016	print fieldata image; TIP first usage creates print file	4144	F15D	labels for TIP PRINT$ files
IOWI$ 024	I/O wait for interrupt	4144	see IOW$; F11A	read/write access checked; GRS initialized
IOXI$ 025	I/O and exit with interrupt	4144	see IOW$; F11A	read/write access checked; GRS initialized
ABSAD$ 030	access to downed memory	4144	F54	Memory must be in 'RV' state; privilege on B1

MCT$ 041	interface to MCT table	4144	F19	QUOTA Level 1; RT$ status needed to write Application Area
READA$ 042	obtain runstream image in ASCII	4144	F3	read access checked
MCORE$ 043	obtain additional main storage	4144	F11D, F11E	memory scrubbed
APRINT$ 060	print ASCII image; TIP first usage creates print file	4144	F15D	labels for TIP PRINT$ files
RT$ 061	raise activity to real time level	4144	F54	QUOTA level 1 control

Intra-TCB: CMS uses to obtain priority memory positioning

APRINT$ 070	ASCII print in standard file	4144	see PRINT$	write access checked
APRNTA$ 071	ASCII print in alternate file	4144	see PRNTA$	write access checked
APNCHA$ 073	ASCII punch in alternate file	4144	see PNCHA$	write access checked
APRTCA$ 076	ASCII alternate print control	4144	see PRNTA$	write access checked
APCHCA$ 077	ASCII alternate punch control	4144	see PNCHA$	write access checked
PFI$ 0104	insert entry in program file	4144	F52	write access checked
PFS$ 0105	search program file TOC	4144	see PFI$	read access checked
PFD$ 0106	delete program file entry	4144	see PFI$	write access checked
PFUWL$ 0107	update next write location in program file	4144	see PFI$	write access checked

PFWL$ 0110	obtain next write location	4144	see PFI$	read access checked
RSI$ 0112	Remote Symbiont Interface Intra-TCB: CMS uses for demand session communication with the Exec	4144	F54	privilege on B1
BDSPT$ 0115	mark a disk track 'bad'	4144	F26	privilege

ER TRMRG$ removed since it is not security relevant

MSCON$ 0125	MFD administration	8486 [6]	F20	security-relevant functions privileged
TSWAP$ 0135	Multi-reel tape volume swap	4144	F63	subsequent reel label access checks
TLBL$ 0142	Tape labeling interface	4144	F3, F8B, F45	Privileged required to read or write VOL1
PRNTA$ 0144	print in alternate file	4144	F53	write access checked
PNCHA$ 0145	punch in alternate file	4144	F53	write access checked
PRTCA$ 0155	print alternate control	4144	see PRNTA$	write access checked
BANK$ 0160	bank manipulation	4144	F2, F11D, F42	reload common bank privileged; residue
ACCNT$ 0163	Quota ACCOUNT$R1 interface	8486	F2, F40	updates by 'Master Account' user-id only
PCHCA$ 0165	punch control alternate	4144	see PNCHA$	write access checked

AREADA$ 0167	ASCII read from alternate file	4144	see READA$	read access checked
SERVE$ 0174	Retrieve Log entries	8486	F54	privilege

ER not used in B1 systems

| USER$ 0177 | TSS$FILE Interface (SIMAN only) | TCB Security Design [7] | F54 | must be requested from SIMAN's common bank |

Intra-TCB: SIMAN uses for TSS$FILE I/O

| SYMB$ 0200 | multi-purpose symbiont ER | 4144 | F15D, F53 | TIP PRINT$ labeling; read/write access checked |

| CKRS$ 0204 | CKPT RESTART | N/A | F37 | Only allowed from CKRS Common Banks; error on B1 |

ER not used in B1 systems

| SREG$ 0213 | create user-id and ACR security records | 11572 | F2, F45 | privilege to create user-id records |

| SUVAL$ 0214 | Security validat. .ı ER | 11572 | F50 | none |

| SUMOD$ 0215 | Security modify user-id records and MFD security records | 11572 | F7, F8A, F13A, F14, F45 | privilege to modify user-id records and change MAC on MFD Ⅰ Ⅰ Ⅰ |

| STAB$ 0216 | set, clear, and retrieve secured ERs and privileges | 11572 | F2, F45 | privilege to set and clear ERs and privileges |

| SDEL$ 0222 | security delete | 11572 | F15C, F45 | privileged except delete of own ACRs |

| SPRNT$ 0223 | security list | 11572 | F18 | privileged except read of own user-id or ACR records |

SABORT$ 0225	unique abort for security reasons	11572	F45	none
DNLOD$ 0226	transfer data between the Exec and the SSP		F54	privilege on B1 This ER is not used in B1 systems
ERCVS$ 0234	Step Control Exec recovery status	8486	F2, F7, F54	privilege on B1

Intra-TCB: IRU & MCB use for processing step control queues

MQF$ 0235	Step Control queue manipulation	8486	F2, F7, F54	privilege on B1

Intra-TCB: IRU, MCB & UDS use for processing step control queues

SC$QR 0236	Step Control update queue item status	8486	F7, F5	privilege on B1

Intra-TCB: IRU uses to recover Step Control data structures

DMABT$ 0237	Step Control DMS abort	8486	F54	privilege on B1

Intra-TCB: UDS uses to inform Step Control of its abort

AUDIT$ 0241	Step Control audit request	8486	F15A-E F25, F54	privilege on B1

Intra-TCB: MCB & UDS use to create audit records

SMOQUE$ 0243	retrieve symbiont queue information retrieval	4144	F2, F12, F48	some functions privileged

Intra-TCB: CMS & PERCON use to process print files

KEYIN$ 0244	retrieve special console keyins	4144	F2, F54	privilege

Intra-TCB: PERCON uses for console interaction (not security relevant)

CONFIG$ 0250	Alter Exec configuration	4144	F2, F39	updates privileged
TRON$ 0263	Common Bank File Assign	8486	F55	Common Bank Subsystems only

MCODE$ 0266	microcode load	4144	F54	privilege
	Intra-TCB: PERCON uses to load printer control units			
IOAID$ 0267	I/O arbitrary device interface	4144	F11C, F43	@ASG of arbitrary device; tape access only on 'J' assign
SYSLOG$ 0272	Create system log entries	4144	F2, F12, F49	privilege, or from a common bank
MODPS$ 0273	modify privileged state	11572	F2, F54	privilege
	Intra-TCB: COMUS, IRU, & PERCON use to activate privileges			
SATTCP$ 0303	Security Object Attribute Compare	11572	F14	none
SCDTL$ 0304	Security - Compartments Table List	11572	F61	User-id Compartment Set only
SCDTA$ 0305	Security - Compartment Table Alter	11572	F9, F44, F2	privilege
TVSLBL$ 0307	Table Volume Security - Label Interface	11572	F3, F8B, F45	Privileged to change MAC; user-id may change DAC
SCLDT$ 0312	Security - Clearance Level Definition table	11572	F2, F46	privilege to update
SCOMCNV$ 0313	Security - Compartments Name Convert	11572	F3, F47	access to user-id set only
RT$INT 02004	Initialize Core Compool	10097 [8]	F54	privilege on B1
	Intra-TCB: CMS uses to initialize TIP environment			
CMS$REG U2006	Register Output Queue with the Exec	10097	F2, F54	privilege on B1
	Intra-TCB: CMS uses to register itself with the Exec			

AC$NIT 02030	Online file initialization	10097	F54	privilege on B1

Intra-TCB: MCB uses this to initialize TIP environment

VT$CHG 02041	VALTAB/VINDEX update	10097	F15A-E F54	privilege on B1

Used by TIP Administrators

VT$PUR 02042	VALTAB load control parameters update	10097	F15, F54	privilege on B1

Used by TIP Administrators

FC$SSN 02043	TIP File I/O	10097	F15C	MAC & DAC checked

TP$APL 02044	TIP Automatic Recovery	10097	F54	privilege on B1

Intra-TCB: MCB uses this to recover the TIP environment

FCREG$ 02045	Exec file registration with TIP	10097	F15, F54	privilege

Used by the TIP Administrator to establish TIP files

DM$IO 02051	File Control emulation of IO$ for DMS	10097	see DM$IOW	MAC & DAC checked

DM$IOW 02052	File Control emulation of IOW$ for DMS	10097	F15C	MAC & DAC checked

TPFLG$ 02062	Flagbox/Logbox ER	10097	F54	privilege on B1

Intra-TCB: MCB uses to set TIP environment

RT$PID 02102	Exec CMS-PID matrix setup	10097	F15, F54	privilege on B1

Intra-TCB: CMS uses to establish TIP environment

SEXEM$ 02103	SEXEM dump	10097	F10, F15D	not allowed for common banks; TIP PRINT$ labeling

TIP$Q 02104   Step Control queue   8486   F15A, privilege on B1
             message   F54
             Intra-TCB: MCB uses to process Step Control queues

QI$CON   Step Control TIP connect 8486   F15B privilege to read
02107   request   other input messages

MCABT$ Step Control MCB abort   8486   F54   privilege on B1
02114

             Intra-TCB: MCB uses to inform Step Control of its abort

FS$UTF   Freespace utility   10097 F54   privilege on B1;
02116   functions   MAC & DAC checked
             Intra-TCB: IRU & MCB use to process TIP file environment

TIP$XMIT   Notify Step Control of   8486   F15,   privilege on B1
02120   Msg Termination   F54
             Intra-TCB: MCB uses to process Step Control message activity

TIP$SM   TIP Session Manager   8486   F15,   must be registered
02130   F54   CMS
             Intra-TCB: CMS & MCB use to process TIP sessions

TIP$TALK   TIP Termination   8486   F15,   must be registered
02131   Communication   F54   CMS
             Intra-TCB: CMS & MCB use to process TIP sessions

SC$SR   IR - recover TIP Session 8486   F7,   privilege on B1
02132   F54
             Intra-TCB: IRU uses to recover TIP session records from audit trail

G.2.4. Other TCB Interfaces

OTHER TCB INTERFACES

Interface	Description	Test #	Security Control
Operator	System console commands	F2, F3, F4, F5, F7, F21	All trusted operation actions are audited

Exec Interrupts	Instructions that interrupt the Exec	F57, F58	Illegal & privileged instructions are handled securely; no memory access is allowed outside user's storage limits
$$ commands	DCP controlled commands	F38	$$ commands perform as documented & $$MSG is disabled
MCB $Action	TIP terminal & commands	F64	Security relevant operator commands $Action commands are restricted to the console operator
LBJ Instructions	Common Bank Subsystem interface	F10, F25, F55, F58, F67	TCB reference monitor must make MAC & DAC checks
Sign-on	Authentication	F9, F15, F56	Valid Compartments, TIP mode, valid passwords, valid clearance levels
TIP File Execute Access	Control TIP Programs	F15C, F15A, F15B F15D, F15E	MAC & DAC
User commands to COMUS	Install non-kernel TCB products, install common banks for testing security controls	F2, F10	system architecture
User commands to FAS	backup/restore files	F4, F14	File labels, LIBLOAD/FAS, MAC & DAC, system architecture
User commands to IRU	establish TIP application environment	F6, F60	system architecture

User commands to SIMAN	establish/modify security records and system security parameters	F56, F22, F23, F24, F27	System architecture, I & A, master account, administrator controls, password handling & validation
User commands to TLABEL	label tape volumes	F6, F8, F14	File labels
User commands to UDS	ADT protection, IRU UDS data base recovery, UDS validation of multiple applications	F25	MAC, system architecture, LBJ

G.3 Specific Hardware Tested

The following systems were used to execute all tests. Each test configuration was validated by the evaluation team.

Component	RT15 (System 11)	S26 (2200/200)	RT9 (1100/90)
IP	4-(K3649-IP) 1-(3065-CAB) 1-(3066-EXP-CAB)	2-(F4111-IP) (F4151-HSMD) 1-(3088-CAB)	1-(3054-03-IP) 2-(F3378-IPPM) 1-(F4088-EXTMON) 4-(1954-IPCU) 3-(8513-MA)
IO	1-(K3652-BMC) 1-(K3650-DCC) 3-(F4257-DCC-II)	2-(K3652-BMC) 2-(F4257-DCC-II) 1-(F4080-IOPEXP) 1-(F3651-BBC) 1-(F3938-PM) 1-(F4078-LBA) 2-(F4079-SHA)	1-(3067-IOP) 3-(K3675-WCM) 3-(K3676-BMCM) 1-(F4077-BPA)
MEM	4-(K3653-HDMSU)	2-(F4073-MEM)	1-(7052-MSU) 2-(K3125-MSU-EXP) 2-(7055-HSPU)

SSP	1-(F3648-SSP)	1-(SSP - included in 3088)	
	1-(8406-diskette unit)		2-(3038-SSP) 2-(3560-SSPCONS) with(F3388-KEYLOCK) 2-(429-CONS-PRINTER)
CONTROLLERS	1-(F3674-ITCU) 1-(F3939-DCPI) 5-(F3163-MSLLM) 1-(F3714-CCU) 1-(3920-CUDSK) 1-(5055-CUTAP) 1-(1974-CAB) 1-(3560-CONS) 1-(429-CONS-PRINTER)	4-(F4113-FRMTR) 1-(5055-CUTAP) 2-(3612-CONS) 1-(429-CONS-PRINTER)	8-(5056/5057-02) 1-(5056-20) 2-(5056-22) 5-(7053-CACHE) 2-(5042-TPCU) 2-(3660-COLOR-CONS) 2-(429-CONS-PRINTER)
DISKS	2-(F4329-DSK) 1-(8451-CAB) 4-(F4640-DSK) 1-(8463-CAB) 3-(8436-DSK)	8-(F4329-DSK) 2-(8451-CAB) 8-(F4115-DSK)	8-(8470-DISK) 12(8480-DISK) 8-(8481-DISK)
TAPES	2-(K3782-STU U18) 2-(884-U26TU) 2-(876-U24TU)	2-(884-U28TU) 1-(884-U26TU)	8-(0874-U36TU)
PRINTERS	1-(789-HSP)	1-(789-HSP)	1-(9246-HSP)
COMMUNICA-TIONS	1-(F3882-FEPI) 1-(2006-ICP) 1-(1986-DCP/10A)	1-(1986-DCP/10A)	1-(1986-DCP/10A)

G.4 Test Coverage

The following table provides more information on what is tested by each test. Identified below are the TCB interfaces whose controls are explicitly tested. Also listed are those interfaces whose correct functioning is implied by the success of the test. Not listed are those interfaces that are commonly-used tools to execute the tests, such as ER RSI$. TCB

INTERFACES TESTED BY TEST

#	TCB INTERFACES TESTED OR USED	Explicit Test of TCB Controls	Implicit Use
F1	TFM - System Generation	TCB Products are built from release versions, with required B1 parameters	
F2	TFM - Install B1 system	B1 is installed according to the TFM	
	SSP - initialize system hardware		Yes
	COMUS	create TCB data structures	
	COMUS - ERs: BANK$, CONFIG$, MODPS$		Yes
	IRU - create TCB data structures		Yes
	CMS1100 - ER CMS$REG		Yes
	IRU - ERs: ERCVS$, MQF$, SC$QR		Yes
	PERCON - ERs: SMOQUE$, KEYIN$, SYSLOG$		Yes
	SIMAN	create TCB data structures	
	SIMAN - ERs: ACCNT$, SCDTA$, SCLDT$, STAB$, SREG$, USER$		Yes
F3	TLABEL	creates tape labels	
	TLABEL - ERs: SCOMCNV$, TLBL$, TVSLBL$		Yes
F4	FAS	save system files	

F5	Save System Security Files		Yes
F6	Audit	Required events, such as console keyins	
	LA - Security Report		Yes
F7	System Architecture	B1 system is recovered	
	IRU - recover IR Application		Yes
	SSP - recover system		Yes
	FAS - ER SUMOD$		Yes
	IRU - ERs: ERCVS$, MQF$, SC$SR, SC$QR		Yes
F8A	Mass Storage @ASG, @FREE, @XQT, @<processor>	DAC & Clearance Level conditions for mass storage files	
	@CAT	correct DAC & CL attributes for mass storage files	
	@LEV		Yes
	ER IOW$	DAC & CL conditions for mass storage files	
	Audit	correct audit records	
	ER SUMOD$	Change file attributes via SIMAN	
F8B	Tape @ASG	DAC & CL conditions for tape files	
	FURPUR Tape ER IOW$	DAC & CL conditions for tape files	
	Audit	correct audit records	

	TLABEL - ERs: TLBL$, TVSLBL$		Yes
F8C	Mass Storage @ASG	Time-of-day DAC	
F9	@ASG	MAC Compartment conditions (mass storage and tapes)	
	@FREE	Compartment file delete conditions	
	@CAT	Correct Compartment attributes	
	Sign-on	Compartment sign-on selection	
	FURPUR - ER IOW$	Compartment file access conditions	
	SIMAN - ER SCDTA$	Privilege & non-privileged	
	Audit	correct audit records	
F10	@ASG	MAC & DAC conditions of common bank subsystems	
	@FREE,D	MAC & DAC file delete conditions of common bank subsystems	
	@START	proper attributes for common bank subsystems	
	ER IOW$	MAC & DAC conditions for common bank subsystems	
	ER SREG$		Yes
	ER SEXEM$	not allowed from common bank	
	LBJ instruction	MAC & DAC conditions	
	Audit	correct audit records	
F11A	@XQT	ensure registers initialized	
	ER FORK$, TFORK$	ensure registers initialized	
	ER IOI$, IOWI$, IOXI$	ensure registers initial. ..1	

F11B	@ASG	ensure file allocations are scrubbed
	ER IOW$ (ACQ$ function)	ensure allocation is scrubbed
F11C	DPREP	ensure DPREP scrubs a disk pack
	Privilege	SSADID required to use DPREP scrub
F11D	@XQT	ensure main storage program allocation is scrubbed
	ER BANK$	ensure main storage allocation is scrubbed
	ER MCORE$	ensure main storage allocation is scrubbed
F11E	TIP transaction	ensure main storage program allocation is scrubbed
	ER MCORE$	ensure main storage allocation is scrubbed
F12	Print Labeling (CMS 1100 & PERCON)	Header/trailer, EPL, sequence numbers
	@CAT	Proper MAC labels
	@SYM	EPL options
	SV & SR Keyins	Print labeling maintained across tape save
	Audit	correct audit records
	PERCON - ERs: KEYIN$, SMOQUE$, SYSLOG$	Yes
	CMS1100 - ERs: SMOQUE$, SYSLOG$	Yes
F13A	@ASG	Unlabeled Disk Packs Security disabling/enabling
	ER SUMOD$	Privilege to clear security disable

	Audit	correct audit records	
F13B	@ASG	Unlabeled Tape Privilege required to @ASG	
F14	FAS	Privileged operations	
	Audit	correct audit records	
	@ASG,CP		Yes I I
	@ASG tape volumes		Yes
	TLABEL		Yes
	ER IOW$		Yes
	FAS - ER SATTCP$	Exec returns proper F-cycle info	
	ER SUMOD$		Yes
F15	All parts use the following:		
	TIP Sign-on		Yes
	CMS 1100 -ERs: RT$PID, TIP$SM, TIP$TALK		Yes
	MCB - ERs: AUDIT$, TIP$XMIT		Yes
	ERs: VT$CHG, VT$PUR		Yes
	ER FCREG$		Yes
	Audit	correct audit records	
F15A	MCB - ER TIP$Q	Public send requires a privilege	
F15B	MCB - ER QI$CON	Retrieval of other's messages requires a privilege	
F15C	ER FC$SSN	MAC & DAC conditions	
	ER DM$IOW	MAC & DAC conditions	

	Execute Access	MAC & DAC conditions	
	ER SDEL$		Yes
F15D	TIP - ER PRINT$	MAC & DAC labels	
F15E	TIP - Sign-on	TIP message access checked	
F16	@@TM	Disabled in B1	
	@SYM & @@SYM @@CONS TM	Disable of @SYM to terminals and userids legal for LIMITED and above console modes	
	@@SEND	not usable in B1	
F17	@FILE	Disabled on B1	
F18	ER SPRNT$	ACR access is controlled	
	Audit	correct audit records	
F19	ER MCT$	access to application area is controlled	
F20	ER MSCON$	DBACK$ & DBACK1$ functions are controlled	
F21	@@CONS	6 console modes controlled	

F22 - F24 are not part of the B1 Test Plan

F25	LBJ	MAC controlled for multi-UDS common bank subsystems	
	UDSC - ADT Interface	Updates controlled	
	Audit	correct audit records	
	UDSC - ER AUDIT$		Yes
	UDSC & IRU Long Recovery	Controlled operation	
F26	ER BDSPT$	requires privilege	

F27 is not part of the B1 Test Plan

F28	ER IOW$	no tape read past EOF or last block written
F29	Hardware	Diagnostic Programs
F30	@XQT	load of NPE programs is disabled
F31	@RUN	userid, account & project-id conditions
F32	@ASG (R, W, and X options)	read-only & exclusive assign conditions
F33	@LEV	privilege usage
F34	@START Audit	'U' & 'Z' options correct audit records
F35	@PMD	User program isolation
F36	@PASSWD & @@PASSWD legal & illegal usage	
F37	@CKPT & @@CKPAR	disabled
	@RSTRT & @RIP	disabled
	ER CKRS$	disabled
F38	$$MSG	disabled
	$$OPEN, $$CLOSE & $$SOFF	require re-validation
	@@TERM	requires re-validation
	@@commands	allowed only at documented session points
	@@device-commands	disabled
F39	ER CONFIG$	change requires privilege
F40	ER ACCNT$	update requires privilege

F41	ER COM$	controlled functions require privilege
F42	ER BANK$	controlled functions require privilege
F43	ER IOAID$	usage requires privilege
F44	ER SCDTA$	controlled functions require privilege
F45	ER SDEL$	use on non-owned objects requires privilege
	ER SREG$	create userid record requires privilege
	ER STAB$	change requires privilege
	ER SUMOD$	controlled functions require privilege
	ER TLBL$	controlled function requires privilege
	ER TVSLBL$	controlled functions require privilege
	ER SABORT$	run is indeed aborted
F46	ER SCLDT$	updates require privilege
F47	ER SCOMCNV$	access only to userid's compartment set
F48	ER SMOQUE$	controlled functions require privilege
F49	ER SYSLOG$	controlled functions requireprivilege
	Audit	correct audit records
F50	ER SUVAL$	provides correct information
F51	@@TOUT$	limited by userid profile
F52	ER PFI$, PFWL$	MAC & DAC conditions
F53	ER SYMB$, PRTCA$, PCHCA$, PNCHA$	MAC & DAC conditions
F54	Controlled ERs	illegal without privilege
F55	ER TRON$	not usable from user program isolation by common bank subsystem

LBJ   Yes

F56	SIMAN commands	validates SIMAN administrator controls, correct menus, batch keyword, valid project & account usage, and valid passwords
F57	Exec Interrupts	Exec correctly handles illegal & privileged instructions
F58	Exec Interrupts	Hardware & Exec correctly handle addressing
	LBJ & LDJ/LIJ	access controlled
F59	@ADD	MAC & DAC read conditions
	@BRKPT	MAC & DAC write
	@SYM	MAC & DAC read & write conditions
F60	IRU commands	administrator operations controlled
F61	ER SCDTL$	access controlled to compartment names
F62	Sign-on	Valid clearance levels
F63	ER TSWAP$	Multi-reel tape volume access checked
F64	MCB $Action Cmds	Restricted to the operator console
F65	Audit	Access controlled and overflow does not lose records
F66	@ASG, @CAT	Correct security label creation
F67	LBJ Instruction	MAC conditions for Common Bank Subsystem Transitions

G.5 Steps to Build and Install the B1 Product

The following table identifies the process of installing a B1 product. Unisys chose to include this process into their test documentation because of the early encounter with the team when Unisys discovered that acceptable testing for an evaluated system could only be performed on evaluated hardware and evaluated software.

Build & Install B1

#	TCSEC Area	Function	Description
F1	Trusted Facility Manual	System Gen	build B1 Exec
F2	Trusted Facility Manual	Install B1	Exec Security Option 2
			Install COMUS & set Private option
			Install 19 products
			Install TCB user-ids, attach AFCB file owners, and set system file security attributes
			Boot Exec Security Option 3
			Secure ERs
			Install MCB & UDS products
F3	Tape Volume Labels	TLABEL	pre-label tapes for LIBSAV
F4	File Labels	LIBSAV	all TCB products
F5	Trusted Facility Manual	Save Security Files	TSS$FILE, SACRD$, ACCOUNT$R1
F6	Audit	Security Log	LA Security Report
F7	System Architecture	recover system	reboot Exec
			Load TSS$FILE & SACRD$
			LIBLOAD TCB products

27 September 1989

G.6 TCSEC Requirements Coverage

This section of tables detail how the B1 Test Plan meets the requirements for the TCSEC class B1. Also, the unique MAC & DAC test conditions are detailed, and the TCB interfaces are mapped against the B1 tests.

G.6.1 TCSEC Test Coverage

The TCSEC outlines the requirements for a B1 system. The table below lists those requirements and states which tests satisfy that requirement.

TCSEC TEST COVERAGE

B1 Section	Title	#	Items Tested	Extent of Coverage
3.1.1.1.	Discretionary Access Control	F8A	DAC for ER IOW$ to mass storage files; also, @XQT, @ER SUMOD$	execute, read, write, assign, modify, and delete access attempts on public, private, and semi-private files (ACRs controlling read, write & execute); file creation defaults to private
		F8B	DAC for ER IOW$ to tape files	read and write access attempts on public, private, and semi-private tape volumes (ACRs controlling read and write); tape volume labels DAC is modified from private (default) to public, and from private to semi-private
		F8C	DAC for ER IOW$ (Time-of-day)	read and write access checks
		F10	DAC for common bank subsystem calls	enter access (execute) to public, private, and semi-private subsystems

F10	DAC for ER IOW$ from subsystem to mass storage files	read, write, assign and delete attempts to public, private, and semi-private files from common bank subsystems.
F15 C	DAC for TIP files	read and write access attempts on public, private, and semi-private TIP files (ACRs controlling read and write); access to semi-private file with deleted ACR also attempted
F15 C	DAC for execute access to TIP programs	execute access attempts on public, private, and semi-private TIP program files (ACRs controlling execute)
F31	Project-id & Account for @RUN	Project-id & account usage controlled
F32	Read-only, write-only, and exclusively assigned files	these accesses function as documented
F52	DAC for ER PFx$	read & write access attempts on semi-private file (ACR controls read and write)
F53	DAC for symbiont ERs	reads and write access attempt on private and semi-private file (ACR controls read and write)
F59	DAC for @ADD, @BRKPT, & @SYM	read and write attempts to public, private and semi-private files (ACR controls reads)

		F63	DAC for multi-reel tape volumes	DAC must be consistent for all reels of multi-reel tape volumes
3.1.1.2.	Object Reuse	F11 A	GRS	proper initialization on all activity starts
		F11 B	Files	proper scrubbing at file allocation
		F11 C	Disk packs	DPREP scrubs
		F11 D	Main memory	proper scrubbing (demand & batch)
		F11 E	Main memory	proper scrubbing (TIP programs)
3.1.1.3.	Labels	F3	Tape volume labels	pre-labels tapes for site admin
		F4	System file labels	maintained across save and restore
		F8A	Clearance levels (mass storage files)	MAC access controls
		F8B	Clearance levels (tape volumes)	MAC access controls
		F9	Compartment sets (mass storage & tape volumes)	MAC access controls
		F10	Subsystem labels	MAC access controls
		F13 A	Labeling imported removable pack files	controlled labeling by authorized user

		F13 B	Unlabeled tape volumes	only privileged user may access unlabeled tape volumes
		F14	Mass storage file labels volumes	maintained across saves & and restores using tape
		F66	Mass storage file labels	proper creation for files created by @ASG or @CAT
3.1.1.3.1	Label Integrity		covered in 3.1.1.3. above	
3.1.1.3.2	Exportation of Labeled		not applicable, since TCB allows no dynamic changes of single-level or multilevel Information communication channels or I/O devices.	
3.1.1.3.2 .1.	Exportation of Multilevel Devices	F3	Backup Tapes	Multilevel system file saves
		F12	Print files	Multilevel labels by PERCON & CMS 1100
		F15 D	Print files	TIP print files labeled
		F13 A	Removable packs	Multilevel files properly labeled
		F14	Backup Tapes	Multilevel tape volume saves and restores
3.1.1.3.2 .2.	Exportation of Single- level Devices	F8B F9	Tape devices	MAC access control checks
		F8A F9	Terminals	MAC security attributes of userid
		F15 E	TIP messages	Security access checks

3.1.1.3.2 .3.	Labeling Human-Readable Output	F12	Print Labels	Sensitivity labels to printers, via PERCON and CMS 1100, top & bottom of every page by default
		F12	Print label override	Auditing of every-page override
		F12	Page separator	Non-spoofable
3.1.1.4.	Mandatory Access Control	F8A	MAC clearance levels for ER IOW$ to mass storage files; also, @XQT, @ER SUMOD$	execute, read, write, assign, modify, and delete access attempts on files
		F8B	MAC clearance levels for ER IOW$ to tape	read and write access attempts on files files
		F9	MAC compartments for ER IOW$ to tape files	read, write, and delete access attempts on files
		F10	MAC for common bank subsystem calls	enter access (execute) to subsystems with system low, single level and multi-level attributes
		F10	MAC for ER IOW$ from subsystem to mass storage files	read, write, assign and delete attempts
		F15 C	MAC for TIP files	read and write access attempts on files
		F15 C	MAC for execute access to TIP	execute access attempts programs
		F52	MAC for ER PFx$	read & write access attempts

		F53	MAC for symbiont ERs	reads and write access attempts
		F59	MAC for @ADD, @BRKPT, & @SYM	read and write attempts
		F63	MAC for multi-reel tape volumes	MAC must be consistent for all reels of multi-reel tape volumes
		F67	MAC for Common Bank Subsystem Transitions	All cases of Untrusted, Single Level Trusted, and Multi Level Trusted.
3.1.2.1.	Identifica-tion and Authentica-tion Project-id	F8A F8B	Userid, Clearance level, Account,	Userid properly identified
		F9	Compartment Set	Userid properly identified
		F10	Subsystem Clearance Level, Compartment Set & Privileges	Common Bank Subsystem properly labeled
		F31	@RUN Userid, Account & Project-id	Userid properly identified & authorized
		F15 A F15 B	TIP Message Privileges	Userid properly identified
		F36	@PASSWD Userid, Password, Clearance Level & Compartment Set	Userid properly identified & authorized
		F38	Userid sign-on	Terminal sign-on controlled
		F40	ACCOUNT$R1	Accounting data protected

		F54	TSS$FILE	Authentication data protected
		F56	Passwords	Authenticate valid password usage
		F56	Passwords	SIMAN never displays passwords
		F56	Account & Project ids	Authenticate valid usage
		F56	SACRD$	SIMAN controls creates and modifies
		F45	SACRD$	Security record creation & modifies via ERs is controlled
		F62	Clearance Level	Authenticate valid usage (Demand/TIP)
		F8A F8B F9, F10 F15A, F15B, F56	Audit	Audit events identify userid security profile
3.1.2.2.	Audit	F6	Audit created and selective reporting	Operator and site administrator actions
		F65	Audit access	Only authorized audit access allowed
		F65	Audit overflow	No loss of audit records possible

(many other tests demonstrate proper auditing and report selection)

3.1.3.1.1	System Architecture	over 30 tests with this connection		
3.1.3.1.2	System Integrity	F29	Hardware Diagnostics	Correct processor and I/O operations

3.1.3.2.   Life-Cycle   No testing requirements
             Assurance

3.1.4.   Documentationl No testing requirements

G.6.2 MAC and DAC Test Conditions

The unique test conditions of MAC & DAC are outlined below. F8A & F8B test conditions 2, 3, 4, & 6.

Note: these conditions assume that the subject (userid) has no privileges.

G.6.2.1 MAC (Clearance Levels)

Clearance Level (6 unique test conditions)

Object Clearance Level	Subject Clearance Level	
	0	Greater than 0
0	1) read, write, execute, delete, enter & modify	2) read & execute
> 0, less than subject	(condition not possible)	3) read & execute (if subject's range includes object's clearance level)
> 0, = to subject	(same as (1))	4) read, write, execute, delete, enter, and modify
> 0, greater than subject	5) no access allowed	6) no access allowed

G.6.2.2 MAC (Compartment Sets)

Test F9 test all 13 conditions.

Compartment Set (13 unique test conditions)

Object Compartment Set	Subject Compartment Set		
	Null	<set>	ALL
Null	1) read, write, execute, delete, enter & modify	2) read & execute	3) read & execute

<set> < than subject	(condition not possible)	4) same as (2)	5) same as (2)
<set> = to subject	6) same as (1))	7) same as (1)	8) same as (1) (object has full set of system compartments)
<set> > than subject	9) no access allowed	10) no access allowed	(condition not possible)
ALL	11) no access allowed	12) no access allowed	13) same as (1)

G.6.2.3 DAC Test Conditions

DAC test conditions are specified with the two tables below. F8A & F8B test conditions 1, 2, 3, 4; F15C tests conditions 5 & 6.

DAC Test Conditions (6 test conditions)

Object DAC Attributes	Subject DAC Attributes	
	Owner	Non-owner
Public	1) read, write, execute, delete, enter & modify	2) same as (1)
Private	3) same as (1)	4) no access allowed
Deleted ACR (same as private)	5) same as (1)	6) no access allowed
Semi-private (ACR attached)	(36 conditions - see table below)	

G6.2.4 ACR Test Conditions

The table below indicates which tests utilize the ACR types.

Note that ACRs attached to userid security records only control 'execute' access.

ACR Test Conditions (36 conditions)

ACCESS SPECIFIED BY ACR	ACR ACCESS TYPES CONTROLLED			
	Read	Execute (Enter)	Write	Delete
Public	F10, F52, F53	F10	F10, F52, F59	
object userid = executing userid	F8A, F8B, F10	F8A, F10 F8B,	F8A, F10, F53	F8A
object userid not = executing userid	F8A, F8B, F10	F8A, F10	F8A, F8B, F10	F8A
object account = executing account	F8A, F8B	F8A	F8A, F8B	F8A
object account not = executing account	F8A, F8B	F8A	F8A, F8B	F8A
object project-id = executing project-id	F8A, F8B	F8A	F8A, F8B	F8A
object project-id not = executing project-id	F8A, F8B	F8A	F8A, F8B	F8A
ACR time within executing time	F8C		F8C	
ACR time not within executing time	F8C		F8C	

G.7 Test Results

This section provides a table which outlines the results of the evaluation team running the Unisys suite of test.

TEST	2200/200	Sys11	1100/90
F1 - F7 Build/Install/Verify B1			
F8A DAC and Clearance Level			R P
F8B DAC and Clearance Levels for Tapes			R P

TEST	2200/200	Sys11	1100/90
F8C DAC by Time of Day	E	E	R P E
F9 Compartments			R P
F10 Common Bank Protection			R P
F11A GRS Residue	Code Reviewed		
F11B Mass Storage Residue		R	
F11C DPREP		R P	
F11D Memory Residue		R P	
F11E Memory Residue (TIP)		R P	
F12 Print Labeling	E	R E	E
F13A Security Disable of Imported Objects	R P		
F13B Unlabeled Tape & EOF			R P
F14 FAS	R P E	E	E
F15A SSTIPSENDMSG Privilege	R P		
F15B SSTIPGETMSG Privilege	R P C		
F15C TIP File Security			R P
F15D TIP Print File Attributes			R
F15E TIP Message Security			R P
F16 Inhibit Terminal Output			R P
F17 @FILE Disable	R P	O P	
F18 ACR Control	R P		
F19 MCT Application Area	R P C		
F20 MSCON$ Privilege	R P C		

27 September 1989

Final Evaluation Report UNISYS OS 1100
Test Descriptions

TEST	2200/200	Sys11	1100/90
F21 Console Privileges	R P C		
F25 UDS Validation, Protection, Recovery			R P
F26 Badspot Privilege		R P	
F29 Hardware Diagnostics		P	
F30 NPE Disable	R P		
F31 @RUN ECL Statement	R P		
F32 @ASG ECL Statement	R P		
F33 @LEV Error Handling	R P		
F34 @START ECL Statement	R P		
F35 @PMD ECL Statement	E	E	R P E
F36 @PASSWD/@@PASSWD Usage		R P C	
F37 CKPT/RSTRT Disabled	R P	O P	
F38 $$ & @@ Commands		R P	
F39 ER CONFIG$ Without Privilege		R P C	O P
F40 ER ACCNT$ Without Privilege		R P C	
F41 SSCONSOLE Privilege	O P	R P	O P
F42 ER BANK$	E	E	R P E
F43 SSADID/ER IOAID$ Privileg		O P	R P
F44 ER SCDTA$ Access	R P C		
F45 Security ER Error Cases	R P		
F46 ER SCLDT$ Access		R P	
F47 ER SCOMCNV$ Access	R P C	O P	

TEST	2200/200	Sys11	1100/90
F48 SSSMOQUE Privilege	R P C		
F49 SSLOGER Privilege	R P E	E	E
F50 ER SUVAL$ File Access	R P C		R P
F51 Terminal Timeout		R P	
F52 Program File Package	R P C		
F53 Symbiont ERs	R P C E	E	E
F54 Controlled ERs	R P E	E	E
F55 ER TRON$			R P
F56 SIMAN and EXEC Userid Controls		R P C	E
F57 Illegal Instructions	R P E	R P E	R P E
F58 Reference Violations & Addressing	R P	R P	R P
F59 Symbiont ECLs (@ADD,@BRKPT,@SYM	R P C		
F60 IRU Userid Without Privileges	R P		
F61 ER SCDTL$ Access			R P
F62 Clearance Level Sign-on	R P C		
F63 Multi-Reel Tape Volumes			R P
F64 MCB Action Commands	R P C		
F65 Audit Access/Overflow/ Recovery	R P C		
F66 Catalogued File Security Label	R P E	E	E
I67	R P		

Legend

R	Run as recommended
P	Pass
C	Comments
O	Optional additional run
E	Test results compared with Unisys test runs

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION **UNCLASSIFIED**		1b RESTRICTIVE MARKINGS			
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION/AVAILABILITY OF REPORT **UNLIMITED DISTRIBUTION**			
2b DECLASSIFICATION/DOWNGRADING SCHEDULE					
4 PERFORMING ORGANIZATION REPORT NUMBER(S) CSC-EPL-89/004		5. MONITORING ORGANIZATION REPORT NUMBER(S) ~~S232,551~~ S 233122			
6a NAME OF PERFORMING ORGANIZATION **National Computer Security Center**	6b OFFICE SYMBOL *(If applicable)* **C12**	7a NAME OF MONITORING ORGANIZATION			
6c ADDRESS *(City, State and ZIP Code)* **9800 Savage Road Ft. George G. Meade, MD 20755-6000**		7b ADDRESS *(City, State and ZIP Code)*			
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL *(If applicable)*	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER			
8c ADDRESS *(City, State and ZIP Code)*		10 SOURCE OF FUNDING NOS			
		PROGRAM ELEMENT NO.	PROJECT NO	TASK NO.	WORK UNIT NO
11 TITLE *(Include Security Classification)* **Final Evaluation Report of UNISYS Corp OS 1100**					
12. PERSONAL AUTHOR(S) Donald G. Crossman, James Donndelinger, Jeffrey Jones, Edward Coyne, Robert Williamson					

13a TYPE OF REPORT **Final**	13b. TIME COVERED FROM ___ TO __	14 DATE OF REPORT *(Yr, Mo., Day)* **890927**	15. PAGE COUNT **187**

16 SUPPLEMENTARY NOTATION

17	COSATI CODES		18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)* **NCSC, TCSEC, Unisys OS 1100**
FIELD	GROUP	SUB GR	

19 ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The security features of the Unisys OS 1100 operating system, configured according to the most secure manner described in the Trusted Facility Manual, running on the 1100/90, the 2200/200, or the System 11 has been evaluated by the National Computer Security Center (NCSC). The security features of OS 1100 were examined against the requirements specified by the *COMPUTER SECURITY SUBSYSTEM INTERPRETATION OF THE DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA (CSSI) dated 16 September 1988.* The NCSC evaluation team has determined that the highest class at which OS 1100 satisfies all the specified requirements of the Criteria is class B1. A system that has been rated as being a B1 class system provides a Trusted Computing Base (TCB) that preserves the integrity of sensitivity labels and uses them to enforce a set of mandatory access control rules.

This report documents the findings of the evaluation of Unisys OS 1100 operating system.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED	21 ABSTRACT SECURITY CLASSIFICATION **UNCLASSIFIED**	
22a NAME OF RESPONSIBLE INDIVIDUAL **DENNIS E. SIRBAUGH**	22b TELEPHONE NUMBER *(Include Area Code)* **(301)859-4458**	8b OFFICE SYMBOL **C12**